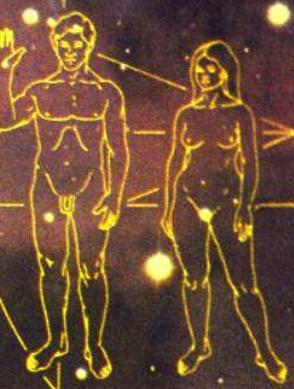
A REVISTA DOS USUÁRIOS DO TK

MICROHOBBY

ANO I - Nº 5 - 1983

Quebra-Cabeça:
MENSAGEM DE VEGA

FITA DO MÊS:
Pulo do Sapo

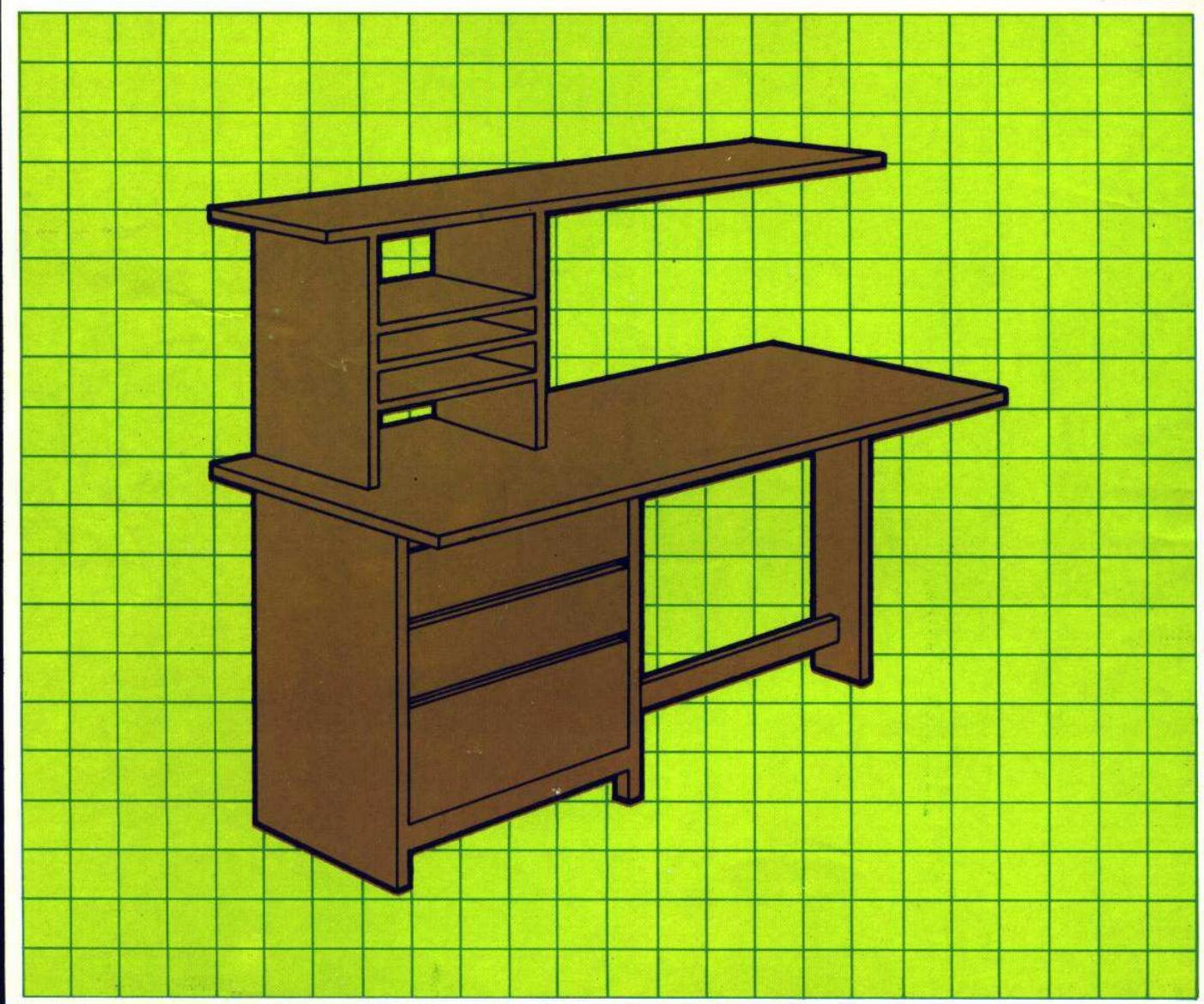
Os oitenta
para compatíveis TRS 80

Teste seus reflexos
Hora sideral
Soma sintática

NOVIDADE:
TK2000 COLOR

micro rack

rack para microcomputador

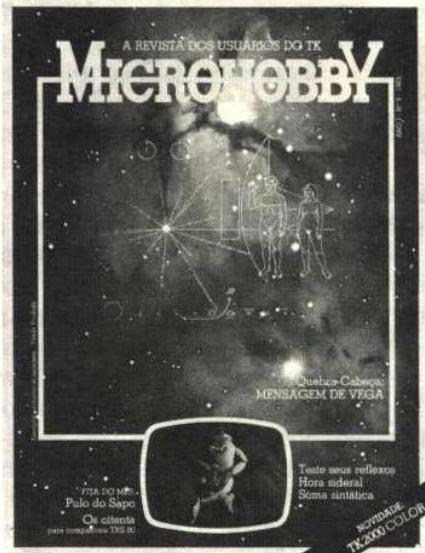
Leoni

- Projetado especificamente para microcomputadores de uso pessoal ou comercial
- Duto eletrificado com tomadas, filtro e circuitos de segurança
- Gaveteiro-arquivo dimensionado para diskettes e fitas cassete
- Dimensões:
  1,30 x 0,65 x 0,65 m e 1,90 x 0,75 x 0,65 m
- Acabamento em madeira natural, laminado ou pintura acrílica.
- Para maiores informações, solicite folheto.

Al. dos Arapanés, 841 - CEP 04524
Tel.: (011) 542-1917 - São Paulo - Brasil

Expediente

DIRETOR-EDITOR:
Pierluigi Piazzi
JORNALISTA RESPONSÁVEL:
Aristides Ribas de Andrade Fº
COORDENAÇÃO EDITORIAL:
Ana Lúcia de Alcântara
ASSESSORIA TÉCNICA:
Flavio Rossini
Wilson José Tucci
ANÁLISE DE SOFTWARE:
Carlos Eduardo Rocha Salvato
Nancy Mitie Ariga
Renato da Silva Oliveira
Roberto Bertini Renzetti
ARTE:
Cassiano Roda
Eliana Santos Queiroz
Fatima M. Rossini Gouveia
Osmère Sarkis
COLABORARAM NESTE NÚMERO:
Enio Eduardo Kranen, Glaucus Brelaz,
Felipe Schmidt, Igor Sartori,
Samuel Ejchel,
Fernando L. Fogliano
e Sinésio Amorim Filho.
CORRESPONDENTES:
Londres — Robert L. Lloyd
Paris — Alain Richard
New York — Natan Portnoy
Milão — Bruno Origo
DEPARTAMENTO ADMINISTRATIVO:
Luiz Carlos Boufelli
Rosana S. Mariano (Assistente)
PUBLICIDADE:
Gerente Comercial:
Gina Elimelek
Criação de Publicidade:
Fernando Leoni
Assistente Comercial:
Atilio Debatin
Secretaria: — Tel.: (011) 256-8348
Rosângela A. Gomes
Contatos:
Aurio J. Mosolino (Supervisor)
Lídia Pauluk, Edson R. Silva
ASSINATURAS: (011) 256-8348
Vendedores:
Carlos Henrique O.S. Carvalho,
Edson Nogueira Brandão,
Ricardo Aparecido Gomes,
Maria Lúcia dos Santos,
Valmirando Oliveira Campos.
Gerente de Assinaturas:
Azarias Cordeiro dos Santos
Secretária: Giselia Vidal Barrasal
MICROHOBBY é editada mensalmente pela MICROMEGA Publicações e Material Didático Ltda. — INPI 2992 — Livro A.
Endereço para correspondência:
Rua Bahia, 1049 — Caixa Postal 60081
CEP 05096 — São Paulo — SP
Tel.: (011) 256-8348. Para solicitar assinaturas (12 números) enviar cheque nominal cruzado à MICROMEGA PMD LTDA., no valor de Cr$ 11.800,00.
Tiragem desta Edição: 60.000 exemplares (número especial para a III Feira Internacional de Informática).
FOTOLITO:
Flash Color
IMPRESSÃO:
Gráfica Castelo
NÚMERO 5:
Distribuição só para assinantes.

COLABORAÇÕES: Os materiais enviados serão submetidos à apreciação e análise por parte da redação e posteriormente remunerados, após sua eventual publicação. As matérias assinadas são de responsabilidade dos autores, inclusive com relação à originalidade das mesmas.

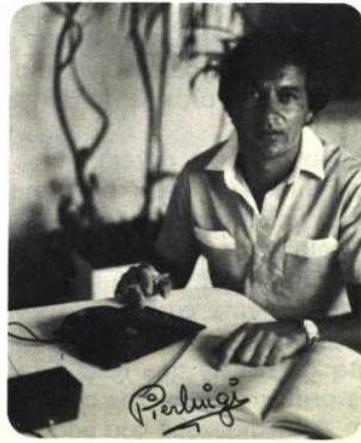
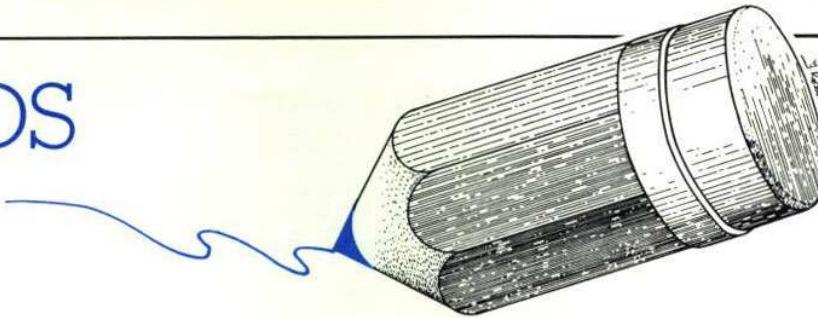
Índice

CAPA:
Hugo Faleiros
Foto da mensagem enviada aos extra-terrestres pela sonda espacial Pioneer, primeiro artefato humano a ultrapassar os limites do sistema solar.

Editorial

A economia brasileira se assemelha a uma escada rolante que está descendo. Se você nela ficar parado, fatalmente acabará descendo junto. O setor ligado à informática, porém está percorrendo esta escada em sentido contrário e consegue galgar seus degraus tão rapidamente que está subindo apesar da crise e da recessão.

Neste mês de outubro realiza-se a III Feira Internacional da Informática no Parque Anhembi em São Paulo. O volume de expositores e participantes mostra bem a importância deste setor da economia brasileira.

Deve-se destacar, entretanto que esta pujança não é devida aos grandes sistemas, área dominada pelos "experts" e pelos pomposos e retumbantes executivos das centrais de processamento de dados. Ela se deve mais ao pequeno empresário ou profissional que se utiliza de um mini ou microcomputador: se deve a você, usuário, que tem uma pequena (e barata!) maravilha dentro de casa. Você deve se orgulhar disto!

Todo este otimismo, porém deve gerar duas cautelas: em primeiro lugar devemos lutar com todas as forças para que esta indústria brasileira continue brasileira! Em segundo lugar devemos nos acautelar contra os picaretas atraídos pelo lucro fácil e pelo modismo do mercado. Para isso a nossa revista continuará informando, analisando e principalmente ***ensinando*** *ao usuário como tirar o máximo proveito de seu pequeno grande computador. Como exemplo disto, estamos publicando neste número a primeira parte de um importante artigo sobre direitos autorais de software. Afinal, muitos de nossos leitores são "pronsumidores", ou seja, consumidores que também produzem em nível artesanal ou até profissional, e esta é uma atividade que pode e deve ser protegida contra piratas e tubarões!*

CARTAS DOS LEITORES

Gostaria de parabenizá-los pela excelente idéia de conglomerar os usuários do TK82C em torno de uma publicação.

Aproveito a oportunidade para solicitar o número zero da **Microhobby** e assim poder completar a coleção, desde o início. A forma de pagamento deverá ser informada pelos senhores, para que eu possa receber o número zero.

Uma vez que a revista tem grande penetração, gostaria de solicitar-lhes, se possível, *a publicação* ou o envio para mim, *do esquema eletrônico do TK-82C, pois tenho vontade de conhecê-lo mais a fundo tanto em* **hardware** *quanto em* **software** *e passar assim, a desenvolver alguns usos diferentes para o micro, principalmente na área de controle.*

Contando com a sua compreensão, agradeço a atenção dispensada.

Luis Henrique P. Fonseca – Salvador, BA.

Caro Luiz Henrique,

Infelizmente o número **Zero** *de nossa revista está esgotado.Como paliativo, porém, você pode tentar obter, de algum amigo, uma cópia "Xerox" dela. Em último caso, escreva-nos novamente e nós lhe mandaremos uma "xerox" da mesma.*

O

Faz pouco mais de três meses, procurei comprar uma máquina de jogos-xadrez e não achei no mercado. Numa das lojas que visitei, me ofereceram um TK-82, dizendo que "também jogava xadrez". O preço do mesmo era razoável e eu o comprei.

Depois de constatar que realmente o computador jogava xadrez, comecei a pensar: "uma máquina capaz de fazer aquilo, deveria ser capaz, também de fazer virtualmente qualquer coisa". Passei a estudar o manual, que a princípio me pareceu escrito em grego, interessando-me cada vez mais pelo computador. Aos poucos, fui me sentindo capaz de desenvolver um programinha, para controle de nosso orçamento doméstico – e não é que deu certo?!

Logo descobri, porém, que a capacidade da máquina não é ilimitada. Não é coisa muito difícil botar, na memória, um programa. Isto fez aumentar minha admiração pelo programa de xadrez. E também minha curiosidade. Quis saber mais a respeito daquela pequena maravilha. E minha curiosidade levou-me a ler, com mais atenção, a revista número **Zero** da **Microhobby.** Gostei tanto da revista que resolvi fazer a sua assinatura. Os dois primeiros números confirmaram a minha expectativa. À medida que vou adquirindo novos conhecimentos, vejo expandir-se o horizonte das possibilidades que a computação oferece.

Enfim: "descobri uma nova atividade extraordinariamente interessante, que tem, a meu ver, um único defeito: não me deixa mais tempo para o xadrez".

Junto com o número 2, recebi a fita brinde, que por algum erro involuntário, de sua parte, veio sem o programa "Simulador de Vôo". Estou remetendo-o para, se possível, trocá-la. E, se for possível escolher, eu prefiro que me enviem o apagador de fitas. Aproveitando o lado 2 da fita, que estava "vazio", gravei nele, uma possível solução do quebra-cabeça "Verdadeiro ou Falso", que deve ser carregado com LOAD "V/F". Mesmo que não seja escolhido para publicação, gostaria que fosse comentado por alguém que realmente saiba programação, para poder corrigir-me e para que eu possa aprender.

Estou enviando, também, uma fita do "TK-MAN" que **não quer entrar.** Gostaria de saber se é defeito de meu gravador ou da fita.

E tenho ainda mais uma pergunta:

1. O manual diz que PAUSE deve ser seguida de POKE 16437,255; mas nos programas publicados na Revista, POKE está **antes** de PAUSE. É o mesmo?
2. Se eu entendi direito, no programa de Alta Resolução (nº 2) há uma linha REM que fornece instruções em linguagem de máquina. As linhas REM não são, então ignoradas? Existe o risco de, ao se utilizar em REM (que pode ser "enfeitado" com alguns caracteres gráficos) estas linhas fornecerem alguma instrução não desejada?

Grato pela atenção que vocês me dispensarem e até a próxima!

Alejandro Luis Cobos – São Paulo, Capital.

Prezado Alejandro,

Grande parte de sua estória é comum a muitos usuários do TK (entre os quais nós nos incluímos). Com relação às fitas, informamo-lhe que a do "Simulador de Vôo" foi substituída por outra testada e já foi providenciado o seu envio para você.

A fita do "TK-MAN" foi remetida à MICROSOFT para ser testada. Oportunamente você receberá notícias dela.

Sua solução do **Quebra-Cabeça** *ainda está sendo estudada por nossos programadores.*

A seguir, damos a resposta de sua primeira pergunta. A resposta da sua segunda pergunta, foi transferida para a seção "DESGRILANDO".

1. A linha POKE 16437,255 após uma PAUSE, só é necessária nas máquinas mais antigas (devido a seu pequeno inconveniente na ROM). As máquinas mais novas dispensam-na.

O

Prezados Senhores,

Antes de mais nada, meus parabéns pela excelente revista; informo-lhes que a **"Microhobby"** é a revista que sempre quis ter em mãos, agora não falta mais nada.

O que quero saber é o seguinte: "A questão de uns quatro meses, adquiri, pela **Microdigital,** a fita com os programas **"Demolidor e Marciano".** E sem pre as usei sem problemas.

Só que de uns dias para cá, o programa **"Marciano"** não **entra** no Micro após o LOAD. Ao invés de acabar com Ø/Ø ele acaba com o cursor **L.**

Eu fiz exatamente o que os itens 1 e 2 falavam na resposta para o **Sérgio** no número anterior, mas não obtive resultado. Logo, eu pergunto: "Seu eu enviar para a **Microdigital** — explicando o problema, eles eventualmente me mandam outra fita?".

E por último, o TK-82C não possue a função "DEF FNR(x)" — o que devo fazer?".

Atenciosamente,

Amauri P. Lúcio — Taubaté, SP.

Caro Amauri,

Cremos que a gravação em fita tenha sido danificada acidentalmente por você (talvez por deixá-la próxima a televisores ou transformadores e outros aparelhos do tipo). Neste caso infelizmente, a fita não lhe será substituída. De qualquer modo, esperando que estejamos errados, vale a pena você tentar enviá-la para verificação. Com relação a garantia das fitas comercializadas, comentamos a respeito, sobre uma inovação introduzida pela **Multisoft** *na seção* **Fita do Mês.**

Você pergunta ainda, como traduzir para o BASIC TK, a instrução DEF FNR (x). A seguir, publicamos, para você, uma possível solução:

Suponha que as linhas a serem traduzidas sejam as seguintes:

```
1000 DEF FNR(x) = LN (INT(x + LNx))
1500 LET K = FNR (10)
```

Procedemos a tradução seguindo os três passos adiante:

1. Colocamos o segundo membro da igualdade da linha 1000 numa STRING.

```
1000 LET F$ = "LN (INT (X + LN (x)))"
```

2. Atribuímos à variável **X** *o valor 10, que é o argumento da função FNR na linha 1500.*

```
1499 LET X = 10
```

3. Atribuímos à variável **K,** *o valor (VAL) da STRING F$.*

```
1500 LET K = VAL (F$)
```

Outras soluções podem ser utilizadas. Como exercício, tente "bolar" uma.

O QUE ESTÁ RESERVADO PARA O Nº 6:

O QUE FOI A FEIRA DA INFORMÁTICA E O CONGRESSO? — Como esteve a Microhobby.

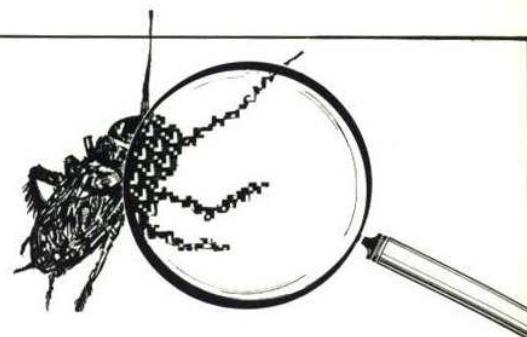
PROGRAMAS DO LEITOR:
Jogos Inteligentes ou Inteligência Artificial
A melhor maneira de escrever programas de jogos inteligentes em BASIC.
Solução de Equação Algébricas e Transcedentes — Através do método "de Bolzano ou de Bissecção".

PROGRAMAS DO MÊS:
Burgo — O programa destaque na edição nº 6.

MUITAS DICAS E INFORMAÇÕES

PEQUENOS ANÚNCIOS — Envie seu pequeno anúncio para ser publicado gratuitamente.

POR DENTRO DO APPLE:
Velocidade e Economia

DESGRILANDO

● *Trecho da Carta de nosso leitor Alejandro Luis Cobos, publicada na seção CARTAS DE NOSSOS LEITORES, neste número:*

. . . Se eu entendi direito, no programa de Alta Resolução (nº 2) há uma linha REM que fornece instruções em linguagem de máquina. As linhas REM não são, então ignoradas? Existe o risco de, ao se utilizar um REM (que pode ser "enfeitado" com algum caractere gráfico), estas fornecerem alguma instrução não desejada?

Aqui está a sua resposta Alejandro:

As linhas **REM** *são ignoradas pelo micro, quando ele as lê em BASIC (dessa forma, não há perigo em "enfeitá-las" com caracteres gráficos). Note que é justamente o fato de serem ignoradas, que as tornam boas lugares para guardarmos programas em linguagens de máquina. Tais programas só serão lidos pelo TK se o instruirmos devidamente, através da função* **USR.**

Para compreender melhor o que dissemos acima, digite o programa listado a seguir, tendo o máximo cuidado para não errar ao digitar os caracteres após o **REM** *da linha 1. Para não haver dúvidas, consulte a TABELA 1, onde eles estão mostrados em seqüência.*

```
   1 REM E£RND7▞-:5? RETURN ?C▀L
EN ▓?7$4 POKE ( LIST TAN
   2 POKE 16522,126
   3 POKE 16524,118
   4▓POKE 16529,119
```

Figura 1

Ao introduzir os caracteres **RETURN, POKE** *e* **LIST** *proceda digitando antes* **SHIFT THEN** *e depois apague os THEN's, movendo o cursor para trás usando, para isso,* **RUBOUT.**

```
ORDEM      OBSERVACAO          CARACTERE

  1     LETRA E                 E
  2     (SHIFT SPACE)           £
  3     (FUNC. RAND)            RND
  4     NUMERO 7                7
  5     (GRAPHICS <>)           ▞
  6     (SHIFT J)               -
  7     (SHIFT Z)               :
  8     NUMERO 5                5
  9     (SHIFT C)               ?
 10     ( THEN RETURN )         RETURN
 11     (SHIFT C)               ?
 12     LETRA C                 C
 13     (GRAF.SHIFT 7)          ▀
 14     (FUNCTION K)            LEN
 15     (GRAF. SPACE)           ▓
 16     (SHIFT C)               ?
 17     NUMERO 7                7
 18     (SHIFT U)               $
 19     NUMERO 4                4
 20     ( THEN POKE )           POKE
 21     (SHIFT I)               (
 22     ( THEN LIST )           LIST
 23     (FUNCTION E)            TAN
```

Figura 2

ATENÇÃO: *Em hipótese alguma traga a linha* **1 REM** *para baixo com o comando* **EDIT.** *Caso você já tenha feito isso, digite* **NEW** *e comece novamente.*

Após haver introduzido a linha 4, digite **RUN** *e* **NEW LINE.** *Quando o Ø/4 surgir na tela, digite* **LIST** *e* **NEW LINE.** *A linha* **1 REM** *. . . deve modificar-se ligeiramente e a listagem deve ser a seguinte:*

```
   1 REM E£RND7▞-:5▓?7$4 POKE (
LIST TAN
   2 POKE 16522,126
   3 POKE 16524,118
   4 POKE 16529,119
```

Figura 3

Se tudo correu bem até aqui, complete o programa como mostrado adiante e ponha-o para rodar:

```
   1 REM E£RND7▞-:5▓?7$4 POKE (
LIST TAN
   2 POKE 16522,126
   3 POKE 16524,118
   4 POKE 16529,119
  10 FOR I=0 TO 255
  20 PRINT CHR$ I;
  30 NEXT I
  40 FOR J=0 TO 40
  50 LET DESGRILANDO=USR 16514
  60 FOR K=0 TO 10
  70 NEXT K
  80 NEXT J
  90 PRINT AT 20,14;"FIM"
```

Figura 4

Observe atentamente o que ocorre na tela. Você digitou **RUN** *e* **NEW LINE** *e o micro começou a ler o programa linha por linha, em BASIC. Vamos analisar como ele lê cada uma delas!*

TABELA II

LINHA	AÇÃO DO TK
1	IGNORA
2	Coloca o número 126 no byte 16522
3	Coloca o número 118 no byte 16524
4	Coloca o número 119 no byte 16529
10 a 30	Imprime o conjunto de caracteres do TK através de um "Looping *FOR-NEXT*"
40	Inicia em "Looping *FOR-NEXT*" de 41 voltas.
50	Ele lê essa linha normalmente até encontrar a função *USR*, cujo argumento é o número 16514. Quando a encontra, pára de ler o programa em BASIC e começa a lê-lo em LINGUAGEM de MÁQUINA, a partir do byte 16514. Acontece que, juntamente nesse byte, encontra-se o código do primeiro caracter após o *REM* da linha 1 (código do caracter E = 42). Prossegue lendo os bytes sucessivos (16515, 16516, etc.) em linguagem de máquina, até encontrar, em um deles, o código 201 correspondente ao caracter *TAN*. Após lê-lo, o micro passa novamente a operar em BASIC, lendo a linha seguinte àquela em que encontrou a função *USR* (linha 60). O caracter *TAN* lido em linguagem de máquina, faz o computador voltar ao BASIC. Essa rotina em linguagem de máquina, é a responsável pela inversão dos caracteres na tela.
60	Fecha o "Looping".
70	Apresenta "FIM" na tela.

Adiante, você pode observar os aspectos da tela após a linha 3Ø e 7Ø serem lidas.

Figura 5

"£$:?()><=+-*
/;,.0123456789ABCDEFGHIJKLMNOPQR
STUVWXYZ

FIM

Figura 6

• Tento participar desta revista expondo algumas dúvidas que tenho. Trata-se da subrotina da ROM do TK-82C, a qual é chamada pela instrução **FF** (RST 38). Meu TK mostra a seguinte seqüência de instruções a partir do endereço ØØ38H:

ODC245OOE1O5C8CBD9ED4FFBE9D1C818F8. . .

Após decrementar "**C**", a rotina salta para o endereço ØØ45 **somente se** a condição for "Não Zero". Então, que sentido tem a instrução do endereço ØØ46H (C8-Ret Z) após a instrução **D1** (POP DE) se, de acordo com vários autores, a instrução "POD DE" não altera flags? Haverá talvez alguma outra parte da ROM que "chame" ou que se "desvie" para o endereço ØØ45H?

Outra coisa: – a instrução **D1** nesta rotina tem outra função que não a de simplesmente retirar (eliminando) o endereço do topo da pilha (stack)?

É possível explicar, mesmo que resumidamente, o funcionamento do registro "R" nesta rotina? E como ocorre o chamamento automático desta sub-rotina durante o processamento do display?

Ainda sobre a ROM do TK, na rotina referente a gravação de programas encontramos as seguintes instruções a partir do endereço 0352H:

DB FE – IN A, (FE)

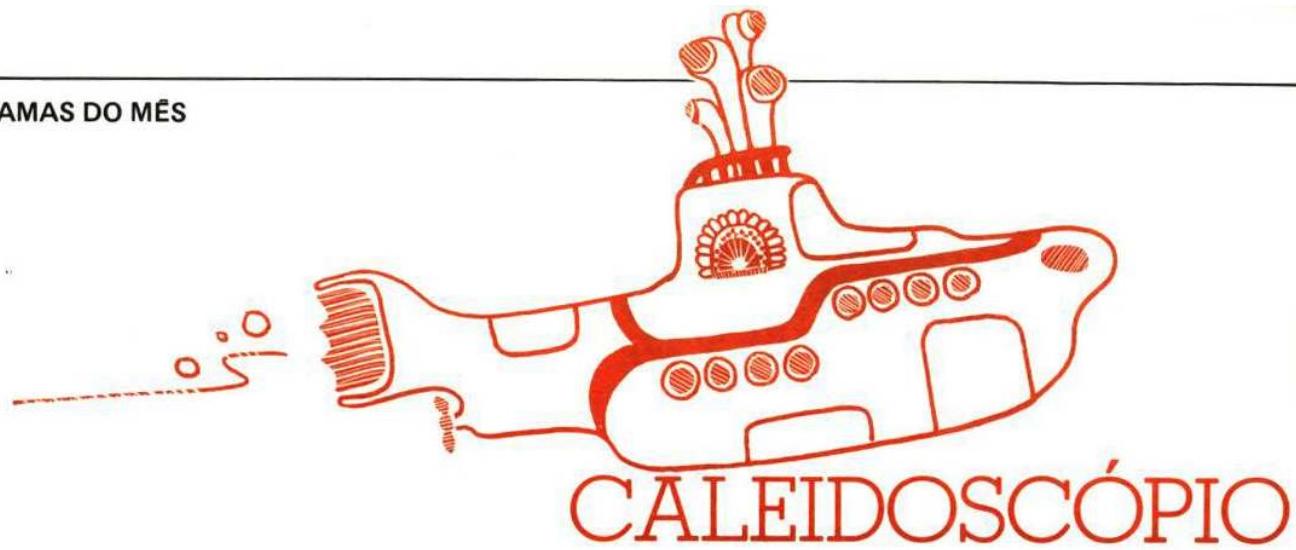
D3 FF – OUT (FF), A

Que efeito tem a instrução OUT **logo após** a instrução **IN,** no caso acima? E a porta **FE,** refere-se ao teclado? Quais as portas de entrada e de saída no TK?

JOSÉ BISIOLI SOBRINHO – São Paulo, SP.

Prezado José,

A instrução POP DE não altera as **flags**, *e nem deve! O RET Z baseia-se no resultado das* **flags** *geradas por instruções* **anteriores** *ao POP DE. Geralmente, as únicas instruções que afetam as* **flags** *convenientemente são as instruções aritméticas e as lógicas (além, é claro, da instrução CP). Com relação às outras dúvidas, por exigirem respostas razoavelmente extensas, sugerimos a leitura dos livros "Introdução à Linguagem de Máquina para o TK" – ASSEMBLY Z8Ø – Vol. I e II –* **Flavio Rossini** *(Editora Moderna e Micromega). O volume I está sendo lançado (na Feira da Informática) e o volume II ainda não tem data prevista de lançamento.*

Infelizmente a maior parte das respostas às suas perguntas encontra-se no volume II.

CALEIDOSCÓPIO

Flavio Rossini

Para simular o efeito de um caleidoscópio na tela do TK, utilizem-se da região central da tela (24 x 24 posições) dividindo-a em quatro partes. Numa das partes será gerado, conforme vocês poderão observar, em posição *aleatória,* um grupo de quatro caracteres iguais (também escolhidos aleatoriamente), os quais devem aparecer imediatamente e simetricamente nas outras três partes. Devido à velocidade requerida, utilizemos então, uma subrotina em ASSEMBLY — que será chamada pelo programa em BASIC — para colocarmos os caracteres na tela.

Comecemos pela subrotina em ASSEMBLY. Ela utilizará dois números gerados pelo BASIC para calcular, a partir do DFILE — posição inicial da tela de TV na memória — onde colocar um caractere *aleatório* no *primeiro* quarto da tela. Este caractere será repetido por 4 vezes, em posições adjacentes e, a seguir, o mesmo grupo será copiado simetricamente nos outros *três* quartos da tela. Note que, ao tratar a tela de TV em linguagem de máquina, não existem *linhas* e *colunas,* mas sim *posições de memória.* Por isso, não estranhe os números gerados pelo BASIC.

Vamos então apresentar a subrotina seguida por um programa em BASIC, o qual permite colocá-la no computador:

MEMÓRIA	MNEMÔNICO	CÓDIGO
16514	—	'ØØ'
16515	—	'ØØ'
16516	—	'ØØ'
16517	—	'ØØ'
16518	LD E, $	'1EØØ'
1652Ø	LD HL, (D-FILE)	'2AØC4Ø'
16523	LD BC, (16514)	'ED4B 82 4Ø'
16527	ADD HL, BC	'Ø9'
16528	LD BC, (16516)	'ED4B844Ø'
16532	SBC HL, BC	'ED42'
16534	CALL TV4	'CDC94Ø'
16537	LD BC, (16516)	'ED4B844Ø'
16541	ADD HL, BC	'Ø9'
16542	ADD HL, BC	'Ø9'
16543	CALL TV4	'CDC9 4Ø'
16546	LD HL, 758	'21F6Ø2'
16549	LD BC, (16514)	'ED4B824Ø'
16553	SBC HL, BC	'ED42'
16555	LD (16514), HL	'22824Ø'
16558	LD HL, (D-FILE)	'2AØC4Ø'
16561	LD BC, (16514)	'ED4B824Ø'
16565	ADD HL, BC	'Ø9'
16566	LD BC, (16516)	'ED4B844Ø'
1657Ø	SBC HL, BC	'ED42'
16572	CALL TV4	'CDC94Ø'
16575	LD BC, (16516)	'ED4B844Ø'
16579	ADD HL, BC	'Ø9'
16580	ADD HL, BC	'Ø9'
16581	CALL TV4	'CDC94Ø'
16584	RET	'C9'
** SUBROTINA TV4 **		
16585	PUSH HL	'E5'
16586	LD (HL), E	'73'
16587	INC HL	'23'
16588	LD (HL), E	'73'
16589	LD BC, 32	'Ø1 2ØØØ'
16592	ADD HL, BC	'Ø9'
16593	LD (HL), E	'73'
16594	INC HL	'23'
16595	LD (HL), E	'73'
16596	POP HL	'E1'
16597	RET	'C9'

OBS.: Na memória 16519, $ representa o caractere que será *enviado* pelo BASIC. As *coordenadas* calculadas pelo BASIC, são colocadas nas memórias de 16514 até 16517.

Este programa tem 84 *bytes* (se incluirmos os dados) e será colocado *dentro* de uma instrução *REM,* que deve estar obrigatoriamente na linha **1.** Assim, comecem a reservar pelo menos 84 **bytes** de memória na linha REM:

```
   1 REM 00000000000000000000000
0000000000000000000000000000000
00000000000000000000000000000
```

(84 ou mais caracteres quaisquer) **Figura 2**

Feito isto, apresentaremos um pequeno programa que lhes possibilitarão colocar a subrotina em ASSEMBLY, dentro do **REM:**

```
   1 REM 000000000000000000000000
00000000000000000000000000000000
0000000000000000000000000000
   5 LET J=16514
  10 LET A$=""
  15 IF A$="" THEN INPUT A$
  20 IF A$="P" THEN STOP
  25 LET X=16*CODE A$+CODE A$(2)
-476
  30 POKE J,X
  35 SCROLL
  40 PRINT "MEM. ";J;"    ";A$( TO
2);"    ";X
  45 LET J=J+1
  50 LET A$=A$(3 TO )
  55 GOTO 15
```

Figura 3

Após colocar o **REM** e o programa acima, no computador, basta teclar **RUN** e **NEW LINE** e começar a digitar os códigos da subrotina em ASSEMBLY, seguidos de **NEW LINE:**

'ØØ' **NEW LINE**
'ØØ' **NEW LINE**
'ØØ' **NEW LINE**
'ØØ' **NEW LINE**
'1EØØ' **NEW LINE**
'2AØC4Ø' **NEW LINE**
"   "
"   "
"   "
'73' **NEW LINE**
'E1' **NEW LINE**
'C9' **NEW LINE**

Neste ponto, a última linha da tela deverá estar indicando:

MEM 16597   C9   2Ø1

Estes caracteres correspondem à memória onde está sendo colocado o código (C9) e seu correspondente decimal (2Ø1). Digitando agora a letra P e **NEW LINE,** o programa pára. Façam **LIST** e **NEW LINE** para verem o que acontece com o **REM** . . .

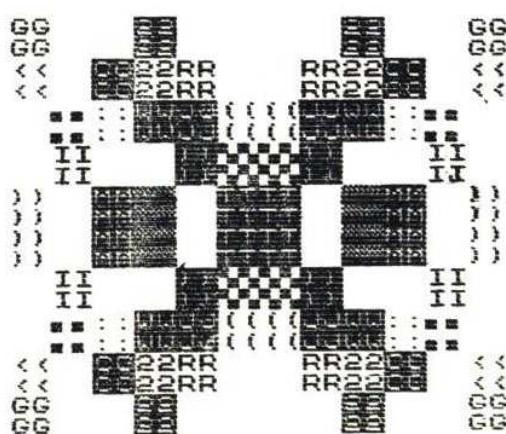
Vamos ver agora, o programa em BASIC que utiliza a subrotina ASSEMBLY para gerar o caleidoscópio. Este programa produzirá em média, um terço de caracteres normais; um terço em vídeo reverso e um terço de espaços em branco, como demonstramos abaixo:

```
   1 REM █ " 2.E£RND GOSUB ?█RND
▒ GOSUB ?█RND GOSUB PILN TAN RND
 GOSUB ?█RND▁▁LN TAN RND5 PLOT ▪
 GOSUB ?█RND GOSUB PI6█RNDE£RND
GOSUB ?█RND▒ GOSUB ?█RND GOSUB P
ILN TAN RND GOSUB ?█RND▁▁LN TAN
RNDTAN FAST ???▪4 ▒??? LPRINT TA
N
   5 REM ** CALEIDOSCOPIO **
  10 REM FLAVIO ROSSINI - 1983
  15 SLOW
  20 RAND
  25 LET X=16+66*INT (6*RND)
  30 LET Y=1+2*INT (6*RND)
  35 POKE 16516,Y
  40 POKE 16514,X-256*INT (X/256
)
  45 POKE 16515,INT (X/256)
  50 LET Z=INT (192*RND)
  55 IF Z>63 AND Z<128 THEN LET
Z=0
  60 POKE 16519,Z
  65 LET A=USR 16518
  70 GOTO 25
```

Figura 4

Execute os programas — aperte **RUN** e **NEW LINE.** Se modificarmos convenientemente as linhas **5Ø** e **55,** poderemos obter um caleidoscópio que utiliza apenas caracteres gráficos:

```
  50 LET Z=INT (17*RND)+128*INT
(2*RND)
  55 IF Z>10 OR Z>138 THEN LET Z
=0
```

Figura 5

Uma outra possibilidade para obtenção do caleidoscópio é reduzir a geração de espaços em branco, suprimindo a linha **55** e alterando a linha **5Ø** para:

```
  50 LET Z=INT (11*RND)+128*INT
(2*RND)
```

Figura 6

Ou ainda para tornar a tela menos carregada, utilizem a seguinte linha:

```
  50 LET Z=INT (10*RND)+1
```

Figura 7

Para tirar uma cópia da tela na impressora basta digitar **"Z"**, após introduzir a linha:

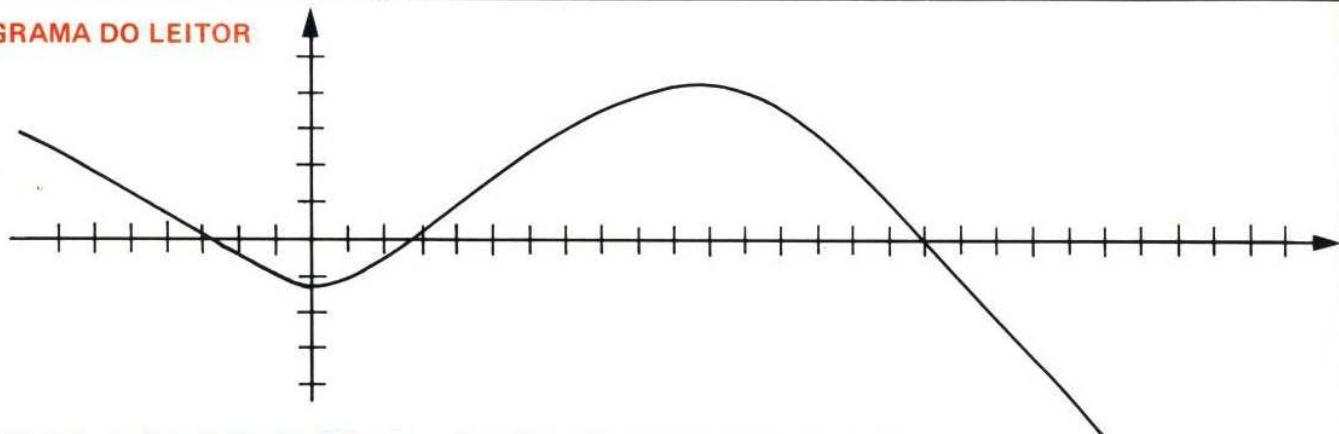
```
  66 IF INKEY$="Z" THEN COPY
```

no programa original.

. . . Picture yourself in a boat on a river,
With tangerine trees and marmalade skies
Somebody calls you, you answer quite slowly
A girl with kaleidoscope eyes . . .

"Lucy in the sky with diamonds"

John Lennon & Paul McCartney - 1967

PROGRAMA DO LEITOR

REGRESSÃO DE ORDEM N

Enio Eduardo Kranen

O programa a seguir tem por finalidade ajustar uma equação, — cuja forma genérica é —:

$$Y = C + A_1 X + A_2 X^2 + \ldots + A_n X^n$$

a um conjunto de pares de valores dados, obtidos na maior parte das vezes por medições de campo ou laboratório.

Na equação dada, temos:

C = Constante
Y = Variável dependente

$A_1, A_2, \ldots, A_n$ = coeficientes da variável independente $X, X^2, \ldots, X^n$.

A seguir, apresentamos um exemplo da variável independente por duas variáveis, sendo **X** a variável independente e **Y** a variável dependente:

X	Y
10,50	23,70
12,80	30,84
17,34	42,01
20,78	47,12
24,11	60,73
27,92	71,35
31,67	82,42
42,63	91,58
50,90	92,33
60,08	95,15

Pede-se para determinar a equação de grau 1 e de grau 2, que ajusta aos pares de valores da tabela.

O programa foi elaborado de forma coloquial, não devendo ocorrer dificuldades no seu uso.

A primeira pergunta indaga sobre o número de casas decimais desejadas na apresentação dos resultados.

Informe, por exemplo, 2.

A seguir informe o número de pares de pontos, no caso 10, introduzindo os mesmos.

Última informação requerida: grau da equação?; informe 1.

```
REGRESSAO DE ORDEM N

NUMERO DE PARES DE PONTOS? 10

X(1)= 10.5        Y(1)= 23.7
X(2)= 12.8        Y(2)= 30.84
X(3)= 17.34       Y(3)= 42.01
X(4)= 20.78       Y(4)= 47.12
X(5)= 24.11       Y(5)= 60.73
X(6)= 27.92       Y(6)= 71.35
X(7)= 31.67       Y(7)= 82.42
X(8)= 42.63       Y(8)= 91.58
X(9)= 50.9        Y(9)= 92.33
X(10)= 60.08      Y(10)= 95.15

GRAU DA EQUACAO?
█
```

Figura 1

Resultados:

```
CONSTANTE:        18.64
COEF 1º GRAU:     1.5

COEF DETERM R2:  0.87

INTERPOLACAO: (ENTRE 0 PARA STOP)

X= 26.12
Y= 58.08

X= 0

V. QUER RODAR OS MESMOS DADOS
COM NOVO GRAU DE EQUACAO (S/N)?

"█"
```

Figura 2

Para permitir pesquisa da equação (grau) que melhor se ajusta aos dados, dotou-se o programa da facilidade de novo processamento, com os mesmos pares de valores **X, Y,** sem a necessidade de reintroduzí-los, o que se torna extremamente vantajoso quando a quantidade de valores for alta. Responda S (sim) à pergunta:

"Você quer rodar os mesmos dados com novo grau de equação (S/N)?"

Grau da equação? 2

Resultados:

```
CONSTANTE:            -18.92
COEF 1º GRAU:         4.31
COEF 2º GRAU:         -.04

COEF DETERM R2:  0.99

INTERPOLACAO: (ENTRE 0 PARA STOP)

X=
```

Figura 3

Nota-se, pela comparação dos dois valores de R2 que a equação de grau 2 se ajusta melhor aos dados, o que também pode ser comprovado traçando-se gráficos das mesmas, com os valores de **X** e **Y**.

O processo acima, poderá ser estendido até graus 5 ou 6 da equação. Lembramos que, à medida que sobe o grau da equação a precisão dos resultados decresce, fato gerado pelo número de casas decimais com que é efetuado o processamento. Não confundir com as casas decimais fixadas anteriormente pelo usuário e que foram utilizadas única e exclusivamente para a apresentação dos resultados e não para o processamento.

A regressão linear é um caso particular da regressão de ordem N, onde o grau da equação foi fixado em 1. (Veja o exemplo anterior).

```
1000 CLS
1010 REM REGRESSAO ORDEM N
1020 LET F$="(INT ((U+0.5/U)*U))
/U"
1030 SCROLL
1040 PRINT "REGRESSAO DE ORDEM N
"
1050 SCROLL
1060 PRINT
1070 SCROLL
1080 PRINT "NUMERO DE PARES DE P
ONTOS? ";
1090 INPUT N
1100 IF N<2 THEN GOTO 1090
1110 PRINT N
1120 SCROLL
1130 PRINT
1140 DIM X(N)
1150 DIM Y(N)
1160 FOR I=1 TO N
1170 SCROLL
1180 PRINT "X(";I;")= ";
1190 INPUT X(I)
1200 PRINT X(I),
1210 IF X(I)<0 THEN GOTO 1210
1220 PRINT "Y(";I;")= ";
1230 INPUT Y(I)
1240 PRINT Y(I)
1250 IF Y(I)<0 THEN GOTO 1280
1260 NEXT I
1270 GOTO 1330
1280 CLS
1290 PRINT
1300 PRINT "AJUSTE SISTEMA DE CO
ORDENADAS    PARA OBTER COORDENAD
AS X E Y     POSITIVAS."
1310 PRINT "DIGITE ""RUN"" PARA
REINICIAR."
1320 STOP
1330 SCROLL
1340 PRINT
1350 SCROLL
1360 PRINT "GRAU DA EQUACAO? ";
1370 INPUT D
1380 PRINT D
1390 IF D<=6 THEN GOTO 1450
1400 SCROLL
1410 PRINT
1420 SCROLL
1430 PRINT "MAIOR GRAU ADMITIDO:
 6"
1440 GOTO 1330
1450 SCROLL
1460 PRINT
1470 SCROLL
1480 PRINT "INFORME O NUMERO DES
EJADO DE"
1490 SCROLL
1500 PRINT "CASAS DECIMAIS NA AP
RESENTACAO";
1510 SCROLL
1520 PRINT "DOS RESULTADOS:
1530 INPUT H
1540 PRINT H
1550 PAUSE 160
1560 CLS
1570 FAST
1580 LET U=10**H
1590 DIM A(2*D+1)
1600 DIM R(D+1,D+2)
1610 DIM T(D+2)
1620 FOR I=1 TO 2*D+1
1630 LET A(I)=0
1640 NEXT I
1650 FOR I=1 TO D+1
1660 FOR J=1 TO D+2
1670 LET R(I,J)=0
1680 NEXT J
1690 NEXT I
1700 FOR I=1 TO D+2
```

continua

```
1710 LET T(I)=0
1720 NEXT I
1730 LET A(1)=N
1740 FOR I=1 TO N
1750 FOR J=2 TO 2*D+1
1760 LET A(J)=A(J)+X(I)**(J-1)
1770 NEXT J
1780 FOR K=1 TO D+1
1790 LET R(K,D+2)=T(K)+Y(I)*X(I)
**(K-1)
1800 LET T(K)=T(K)+Y(I)*X(I)**(K
-1)
1810 NEXT K
1820 LET T(D+2)=T(D+2)+Y(I)**2
1830 NEXT I
1840 FOR J=1 TO D+1
1850 FOR K=1 TO D+1
1860 LET R(J,K)=A(J+K-1)
1870 NEXT K
1880 NEXT J
1890 FOR J=1 TO D+1
1900 FOR K=J TO D+1
1910 IF R(K,J)<>0 THEN GOTO 1970
1920 NEXT K
1930 SLOW
1940 SCROLL
1950 PRINT "NAO HA SOLUCAO UNICA
"
1960 GOTO 2610
1970 FOR I=1 TO D+2
1980 LET S=R(J,I)
1990 LET R(J,I)=R(K,I)
2000 LET R(K,I)=S
2010 NEXT I
2020 LET Z=1/R(J,J)
2030 FOR I=1 TO D+2
2040 LET R(J,I)=Z*R(J,I)
2050 NEXT I
2060 FOR K=1 TO D+1
2070 IF K=J THEN GOTO 2120
2080 LET Z=-R(K,J)
2090 FOR I=1 TO D+2
2100 LET R(K.I)=R(K,I)+Z*R(J,I)
2110 NEXT I
2120 NEXT K
2130 NEXT J
2140 SLOW
2150 SCROLL
2160 PRINT
2170 SCROLL
2180 PRINT "CONSTANTE:",
2190 LET W=R(1,D+2)
2200 PRINT VAL F$
2210 FOR J=1 TO D
2220 SCROLL
2230 PRINT "COEF ";J;"■ GRAU:"
2240 LET W=R(J+1,D+2)
2250 PRINT VAL F$
2260 NEXT J
2270 SCROLL
2280 PRINT
2290 LET P=0
2300 FOR J=2 TO D+1
2310 LET P=P+R(J,D+2)*(T(J)-A(J)
*T(1)/N)
2320 NEXT J
2330 LET Q=T(D+2)-T(1)**2/N
2340 LET Z=Q-P
2350 LET I=N-D-1
2360 SCROLL
2370 PRINT
2380 LET W=P/Q
2390 SCROLL
2400 PRINT "COEF DETERM R2:",VAL
 F$
2410 SCROLL
2420 PRINT
2430 SCROLL
2440 PRINT "INTERPOLACAO: (ENTRE
0 PARA STOP)"
2450 SCROLL
2460 PRINT
2470 LET W=R(1,D+2)
2480 SCROLL
2490 PRINT "X= "
2500 INPUT X
2510 PRINT X
2520 IF X=0 THEN GOTO 2610
2530 FOR J=1 TO D
2540 LET W=W+R(J+1,D+2)*X**J
2550 NEXT J
2560 SCROLL
2570 PRINT "Y= ";VAL F$
2580 SCROLL
2590 PRINT
2600 GOTO 2470
2610 SCROLL
2620 PRINT
2630 SCROLL
2640 PRINT "V. QUER RODAR OS MES
MOS DADOS"
2650 SCROLL
2660 PRINT "COM NOVO GRAU DE EQU
ACAO (S/N)?"
2670 SCROLL
2680 PRINT
2690 INPUT Q$
2700 IF Q$="S" OR Q$="SIM" THEN
GOTO 1350
2710 SCROLL
2720 PRINT "V. QUER RODAR O PROG
RAMA COM"
2730 SCROLL
2740 PRINT "NOVOS DADOS (S/N)?"
2750 INPUT Q$
2760 IF Q$="S" OR Q$="SIM" THEN
RUN
2770 STOP
2780 SAVE "REGRESSAO DE ORDEM ■"
2790 RUN
2800 STOP
```

HORA SIDERAL

Fernando L. Fogliano

Localizar astros, nebulosas, galáxias ou qualquer objeto no firmamento, nem sempre é uma tarefa fácil para os astrônomos amadores. Às vezes, por não estarem bem familiarizados no reconhecimento das constelações e em outras devido à pequena luminosidade destes astros, o que dificulta ou até impede a sua localização visual.

Este programa é uma útil ferramenta quando se deseja localizar qualquer ponto na esfera celeste fornecendo, uma dada hora solar, à hora sideral correspondente.

Para aqueles que não estejam bem familiarizados com os elementos usados no cálculo, bem como sua terminologia, aqui vão algumas explanações:

O tempo necessário para o Sol cruzar 2 vezes consecutivas um determinado meridiano é chamado "Dia Solar", que é constituído, por convenção, de 24 horas solares.

Quando o Sol está sobre o nosso meridiano seu ângulo horário, H, é nulo e nosso relógio marca 12 horas ou meio dia. Matematicamente podemos escrever:

$$t = H + 12$$

onde:

t = tempo solar local
H = ângulo horário do Sol medido em horas solares.

Em analogia à hora solar pode-se definir a Hora Sideral. O dia sideral é o intervalo de tempo decorrido entre duas passagens consecutivas de qualquer estrela por um determinado meridiano e, tem por convenção 24 horas siderais. Mas os astrônomos não usam, qualquer estrela para medir a Hora Sideral. Existe um ponto fictício chamado Ponto Vernal ou Ponto γ (gama) e, é através do ângulo horário deste ponto que se mede a Hora Sideral, com uma diferença em relação ao tempo solar; não se adicionam 12 horas ao ângulo horário do ponto vernal. Matematicamente podemos escrever:

$$T = H$$

onde:

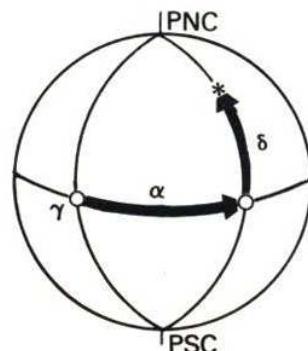
T = tempo sideral
H = ângulo horário do ponto vernal medido em horas siderais.

É fácil constatar-se que o **dia sideral** é cerca de 4 minutos menor que o **dia solar.** Desta forma, o **ponto vernal** e todas as demais estrelas cruzam o meridiano de um determinado local aproximadamente 4 minutos mais cedo que no dia anterior (movimento anual da esfera celeste).

Para se localizar um ponto qualquer na esfera celeste, os astrônomos criaram um sistema semelhante ao sistema de coordenadas geográficas (longitude e latitude). É o chamado **Sistema Equatorial de Referências.** Neste sistema o astro possui, analogamente ao sistema geográfico, duas coordenadas, a saber: **Ascenção Reta** e **Declinação.** Para melhor se compreender estes elementos, podemos fazer uma analogia entre **Ascenção Reta** e Longitude Geográfica e, **Declinação** e Latitude Geográfica. Vamos definir então, o que significam estas coordenadas:

Ascensão Reta α — É a distância em horas siderais, contadas a partir do **ponto vernal,** sobre o Equador Celeste, ao **Círculo Horário** de um astro qualquer no sentido oeste-leste. Varia de Ø a 24 horas.

Desta definição vemos que o **ponto vernal** é a origem das ascensões retas, daí a necessidade de localizá-lo calculando a **Hora Sideral.**

Declinação δ — É a distância em graus, medidos sobre o Círculo Horário, a partir do equador a um determinado astro. A declinação varia de Ø° a 9Ø° Norte e de Ø a -9Ø° ao Sul.

Dados de alimentação do micro-computador

1. **Hora Sideral a Ø Hora de Grenwich — Tempo Universal HSØ** — Este valor pode ser encontrado no **Anuário**

Astronômico do Instituto Astronômico e **Geofísico da Universidade de São Paulo,** na seção **Sol,** sob o título "A Øh T.U. HORA SIDERAL". Este valor nos fornece a posição do **ponto vernal,** quando for Ø Hora em Greenwich.

2. **Longitude** — É uma das coordenadas geográficas do ponto onde se dá a observação. É dada em horas, minutos e segundos. Para que se faça a conversão de graus em horas, basta relacionar: 24 horas — 360 graus; 1 hora — 15 graus.

3. **Fuso Horário** — É o fuso horário no qual nos encontramos, positivo a oeste de Greenwich e negativo a leste.

4. **Hora Local** — É a hora solar na qual estamos querendo calcular a hora sideral.

Para São Paulo, a longitude é aproximadamente três horas, 6 minutos e 30 segundos e o fuso horário é de três horas; valor este que vale para toda a faixa Atlântica brasileira.

```
  10 PRINT AT 1,1;"        [illegible]
[illegible]"
  20 PRINT AT 3,1;"       INTRODUZA
A HORA SIDERAL À        0  HORA DE
 TEMPO UNIVERSAL:"
  30 GOSUB 300
  40 LET HSDEC=DEC
  50 PRINT AT 3,1;"INTRODUZA A L
ONGITUDE"
  60 GOSUB 300
  70 LET LDEC=DEC
  80 PRINT AT 3,1;"INTRODUZA O F
USO HORARIO"
  90 INPUT FUSO
 110 CLS
 120 PRINT AT 3,1;"INTRODUZA A H
ORA LOCAL"
 130 GOSUB 300
 140 LET HLDEC=DEC
 150 LET TU=HLDEC+FUSO
 160 LET DELTS=TU*366.25/365.25
 170 LET TSG=HSDEC+DELTS
 180 IF TSG>24 THEN LET TSG=TSG-
24
 190 IF TSG<0 THEN LET TSG=TSG+2
4
 200 LET TSL=TSG-LDEC
 210 LET A=(TSL-INT (TSL))*60
 220 LET B=(A-INT (A))*60
 230 SLOW
 240 PRINT AT 10,1;"A HORA SIDER
AL LOCAL E:";AT 12,3;INT (TSL);
AT 12,13;"HORAS";AT 14,3;INT (A)
;AT 14,13;"MINUTOS";AT 16,3;B;AT
 16,13;"SEGUNDOS"
 250 PRINT AT 18,0;"VOCE QUER CA
LCULAR A HORA      SIDERAL LOC
AL  PARA  OUTRO       INSTANTE?(S/
N)"
 270 IF INKEY$="S" THEN GOTO 100
 280 IF INKEY$="N" THEN STOP
 290 GOTO 270
 300 FAST
 310 PRINT AT 6,1;"QUAL A HORA?"
 320 INPUT H
 330 PRINT AT 7,4;H
 340 PRINT AT 8,1;"QUANTOS MINUT
OS?"
 350 INPUT M
 360 PRINT AT 9,4;M
 370 PRINT AT 10,1;"QUANTOS SEGU
NDOS?"
 380 INPUT S
 390 PRINT AT 11,4;S
 400 LET D=M+S/60
 410 LET E=D/60
 420 LET DEC=H+E
 430 PAUSE 66
 440 CLS
 450 RETURN
```

aula 5

Flavio Rossini

Nesta aula, iremos introduzir um novo tipo de raciocínio usado para elaborar programas: trata-se de um raciocínio GRÁFICO que facilita a "visualização" de um programa e cuja utilidade torna-se evidente" quanto mais complexo for o programa a ser realizado.

Este raciocínio é denominado ***FLUXOGRAMA*** ou ***DIAGRAMA DE BLOCOS*** e baseia-se no fato de associar, a cada idéia básica usada em programação, um símbolo; sendo que, os vários símbolos são interligados por linhas orientadas, que indicam o fluxo de execução do programa. Outra vantagem deste método é a sua universalidade; de fato, como as idéias básicas de qualquer linguagem são as mesmas, uma vez feito um programa em diagrama de blocos, ele poderá ser facilmente "traduzido" para BASIC, FORTRAN, ASSEMBLY, COBOL ou qualquer outra linguagem, desde que ela não esteja estruturada (o significado deste termo não pode ser explicado num curso introdutório como este).

Até o presente, as idéias básicas vistas por nós foram:

SÍMBOLO

a. Entrada de dados por teclado:

(BASIC = INPUT)

Fig. 1

b. Saída de dados para a tela:

(BASIC = PRINT)

Fig. 2

c. Loops:

(BASIC = FOR/NEXT)

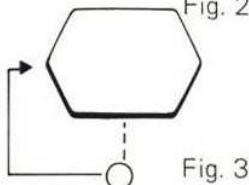



Fig. 3

d. Instruções matemáticas utilizando variáveis de memória e comandos diretos:

(BASIC = LET/CLS/RAND/ /+/–/Etc.)

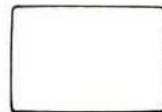



Fig. 4

e. Início de programa: (I)

Fig. 5

f. Fim de programa: (F)

Fig. 6

g. Continuação do fluxograma, quando não há espaço físico:

Fig. 7

* qualquer símbolo sem ser **I, F** ou **R.**

Vamos apresentar agora um outro conceito, talvez o mais importante em programação: a tomada de decisões por parte do computador! Esta é uma ordem que torna o computador capaz de decidir — baseado numa condição específica — qual "caminho" seguir na execução de um programa. Esta característica é que faz com que o computador seja chamado de "máquina inteligente". O símbolo utilizado para tomada de decisões é o da figura seguinte:

Fig. 8

Para exemplificar, façamos um programa que "pergunte" um número, coloque-o na variável **N** e escreva na tela BELEZA, se o número for maior do que **100** e ALEGRIA, em caso contrário. Este seria o FLUXOGRAMA:

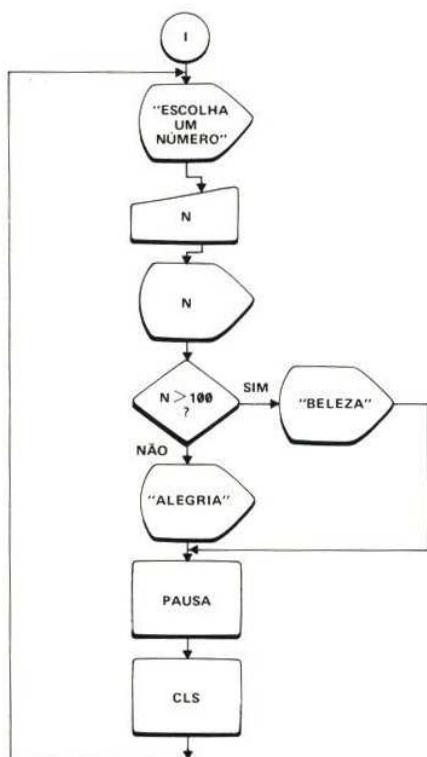



Fig. 9

Note que a idéia de tomada de decisão, ao contrário das demais, implica em uma "entrada" e pelo menos "duas saídas" (todas as outras têm apenas uma **entrada** e uma **saída**). Surge assim, uma pequena dificuldade ao tentar passar este programa para o BASIC, pois as instruções são escritas sequencialmente, ou seja, numa única "dimensão". A instrução de decisão tem a seguinte forma:

IF	CONDIÇÃO	THEN	FAÇA ALGO

Fig. 10

Caso contrário, prossiga para a próxima instrução. A **key-word IF** está na tecla **U** e o **THEN** na tecla 3. Note que esta estrutura é a única que permite o aparecimento de **duas key-words** numa única instrução; de fato, após o **THEN,** o cursor volta a indicar **K**.

Vamos então ao nosso programa:

```
 10 PRINT "ESCOLHA UM NUMERO",
 20 INPUT N
 30 PRINT N
 40 PRINT
 50 IF N>100 THEN GOTO 100
 60 PRINT "ALEGRIA"
 70 PAUSE 150
 80 CLS
 90 GOTO 10
100 PRINT "BELEZA"
110 GOTO 70
```

Fig. 11

Na linha **50** o **TK** toma a decisão: se **N > 100;** caso contrário, prossegue para a linha **60.** (O símbolo > está na tecla **M**).

Perceba como é mais simples normalizar o programa em diagrama de blocos. Note que o **GOTO** não tem símbolo especial: ele corresponde à uma linha orientada. Vale a pena salientar também que não é necessário incluir detalhes nos fluxogramas: por exemplo, no programa anterior o comando de pular linha, não foi incluído. Apenas as idéias principais devem ser "desenhadas"; isto facilita a visualização.

Antes de continuar com mais exemplos de utilização do fluxograma, vamos apresentar mais duas instruções. O TK tem duas velocidades de trabalho: uma lenta (**SLOW**) e outra rápida (**FAST**). Quando ele opera em **SLOW,** ele se preocupa em executar o programa e, ao mesmo tempo, mostrar os resultados na tela; em **FAST,** ele "esquece" a tela e os resultados só são apresentados durante uma PAUSA, ou no final do programa. Naturalmente, cada uma tem suas vantagens e tudo depende de cada caso. Quando o **TK** é ligado ele automaticamente opera em **SLOW**. No entanto, para digitar um programa é mais conveniente usar a velocidade mais rápida; de fato experimente digitar:

```
LET X=12345678901234567890
```

Fig. 12

A seguir, digite **FAST** (**shift F**) e **NEW LINE** e repita a operação acima: (OBS.: Para entender bem estes comandos, execute os programas dados na aula 3 em **SLOW** e em **FAST**).

Vamos então fazer um programa que "chuta" um número; você terá que adivinhá-lo no menor número de tentativas possíveis.

Figura 13

continua

O que, em BASIC, ficaria:

```
  10 RAND
  15 LET X=INT (10*RND)+1
  20 PRINT "ADIVINHE UM NUMERO D
E 1 A 10 QUEEU VOU CHUTAR"
  30 LET I=1
  40 INPUT A
  50 CLS
  60 IF A>X THEN PRINT "O NUM. Q
UE EU CHUTEI E < DO QUE ";A
  70 IF A<X THEN PRINT "O NUM. Q
UE EU CHUTEI E > DO QUE ";A
  80 IF A<>X THEN GOTO 100
  90 GOTO 130
 100 PRINT "TENTE OUTRA VEZ"
 110 LET I=I+1
 120 GOTO 40
 130 PRINT "VOCE ACERTOU EM ";I;
" TENTATIVAS"
```

Execute o programa! Perceba o uso da variável I como "contador" de tentativas. O símbolo < > está na tecla T e o símbolo < está na tecla N.

Este programa poderia ser escrito mais "elegantemente", eliminando a linha 90 e substituindo a linha **80** por:

```
80 IF A=X THEN GOTO 130
```

Figura 14

CABO COAXIAL
OU ANTENA ¼ e ½ ONDA.

Sinésio Amorim Filho

Este é um programa simples, compatível aos micros de 1K byte , onde o radioamador ou projetistas de antenas podem calcular rapidamente o múltiplo de onda para cálculo do comprimento do cabo coaxial a ser usado em uma estação transceptora e até mesmo o comprimento de uma antena para 1/4 ou 1/2 onda. É um programa de acesso rápido, onde a máquina questiona o operador sobre qual o serviço a ser executado — utilizando-se de uma linguagem comum, e lhe responde resultando deste diálogo, um bate-papo entre operador e máquina.

Logo após você ter passado o programa, pressione o **RUN** e **NEW LINE** e aparecerá um cabeçalho do programa, seguido da linha abaixo de uma pergunta: "Você quer o cálculo do cabo ou antena?". Dependendo de sua necessidade, digite um ou outro, mas observe o seguinte: se você digitar algo diferente de cabo ou antena, a máquina rejeitará e imprimirá o seguinte: "Você digitou errado, digite CABO ou ANTENA". Num determinado passo do trabalho, a máquina perguntará a freqüência a ser calculada, na qual deverá ser digitada apenas a freqüência direta, pois a máquina já está programada para trabalhar em MHz. Exemplo: Para dois metros, deverá ser digitado apenas 144 ou no caso, uma freqüência específica 144.75. Se você digitar 144000 a máquina entenderá 144Ghz.

Observação: Após ter digitado qualquer resposta ao computador, sempre pressione **NEW LINE.**

Agora Mãos à Obra . . .

```
   5 REM "SINESIO AMORIM FILHO 1
983"
  10 PRINT "[illegible]"
  15 PRINT
  20 PRINT
  30 PRINT
  40 PRINT "          -VOCE QUER O C
ALCULO?O-            DO CABO OU A
NTENA?"
  50 INPUT A$
  55 PRINT
  56 CLS
  60 IF A$="CABO" THEN GOTO 100
  65 IF A$="ANTENA" THEN GOTO 25
0
  70 IF A$<>"CABO,ANTENA" THEN G
OTO 200
  75 CLS
  76 PRINT
  77 PRINT
 100 PRINT "OK...VOCE QUER O CAL
CULO DO CABO"
 101 PRINT
 102 PRINT
 110 PRINT "QUAL A FREQUENCIA?"
 120 INPUT X
 121 CLS
 130 PRINT "-PARA A FREQUENCIA D
E ";X;"MHZ..."
 135 LET W=300/X
 140 LET Z=W*.66
 145 LET P=Z/4
 148 PRINT
 149 PRINT
 150 PRINT
 155 PRINT "▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄"
 160 PRINT "...O FATOR E ";P;" M
ETROS"
 165 PRINT "▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄"
 166 PRINT
 167 PRINT
 168 PRINT
 169 PRINT "     QUER MAIS ALGUM C
ALCULO?"
 170 INPUT B$
 171 CLS
 175 IF B$="SIM" THEN GOTO 40
 180 IF B$="NAO" THEN GOTO 185
 181 PRINT
 182 PRINT
 183 PRINT
 185 PRINT "       ESPERO TER AJUD
ADO?
 186 PRINT
 187 PRINT
 188 PRINT
 189 PRINT "              TCHAU..."
 190 STOP
 200 PRINT "VOCE DIGITOU ERRADO,
DIGITE               CABO OU ANTEN
A"
 210 PRINT "      ▄▄▄▄   ▄▄▄▄▄
▄"
 215 GOTO 50
 250 PRINT "OK...VOCE QUER ANTEN
A"
 253 PRINT
 254 PRINT
 255 PRINT "QUAL A FREQUENCIA?"
 260 INPUT Y
 265 CLS
 270 LET R=142.5/Y
 275 LET U=R/2
 277 PRINT
 278 PRINT
 279 PRINT "PARA A FREQUENCIA DE
";Y;"MHZ..."
 283 PRINT
 284 PRINT
 285 PRINT
 290 PRINT "▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄"
 295 PRINT "O COMPRIMENTO E DE "
;U;" METROS PARA 1/4 DE ONDA"
 300 PRINT "▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄"
 310 PRINT "▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄"
 315 PRINT "E O COMPRIMENTO E DE
";R;" METROS PARA 1/2 ONDA"
 320 PRINT "▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄"
 344 PRINT
 350 GOTO 169
 355 SAVE "CALCUL█"
 360 RUN
```

PROGRAMA DO MÊS

SOMA SINTÁTICA

Glaucus Brelaz

Ao se teclar um programa em um TK82, 83 ou 85 e após carregá-lo em fita, sempre resta ao usuário uma indagação: "Como saber se todo o programa foi teclado corretamente ou se a transferência da fita para o computador foi perfeita?" . . . A verdade é a seguinte: Mesmo que apenas um caractere seja confundido ou apenas um **bit** alterado durante a carga, estes são motivos suficientes para que o programa seja executado com erro, ou até mesmo não rode na máquina.

A forma que propomos para conferir o conteúdo da memória é a **Soma Sintática** (efetuada após o programa ter sido teclado ou carregado da fita).

A **Soma Sintática,** consiste na adição dos valores de todos os **bytes** de um programa que fornecerá um total tal, a ser checado todas as vezes que o programa for carregado. Nos nossos programas, a partir do número seis de **Microhobby** — passaremos a informar a **Soma Sintática** de cada programa, que deverá ser conferido com o valor fornecido pelo seu computador.

Como o programa que executa a **Soma Sintática** não deve alterar o conteúdo do seu programa original, optamos por fazê-lo em **linguagem de máquina,** o que nos dá ainda a vantagem de uma maior rapidez de execução. Ademais, impomos três condições para a execução de um programa:

1. O programa não poderá estar em uma sentença **REM,** pois muitos programas em BASIC, utilizam subrotinas em máquina no **REM** (Endereço 16 514).

2. O programa não estando na área de memória destinada ao **BASIC,** deve ser providenciada uma maneira de salvá-lo em cassete.

3. O programa deverá ser recolocado na memória, de forma a ser utilizado em **2, 16 ou 64 K RAM.**

Decidimos então, colocar o programa dentro de uma sentença **REM,** transferindo-o para uma localização de memória escolhida acima da área reservada ao BASIC.

Para criar sua fita com a função **Soma Sintática,** você deve executar os seguintes passos:

1. Tecle o programa da **listagem 1** para carga de código hexadecimal.

LISTAGEM - 1

```
  10 REM 00000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
0000000
  20 LET X$=""
  30 LET X=16514
  40 IF X$="" THEN INPUT X$
  50 IF X$(1)="S" THEN STOP
  60 PRINT X$(1 TO 2);" ";
  70 POKE X,(16*CODE X$+CODE X$(
2)-476)
  80 LET X=X+1
  90 LET X$=X$(3 TO )
 100 GOTO 40
```

2. Execute o programa dando entrada no código hexadecimal da listagem 2.

```
                    LISTAGEM - 2
→
CD   23   0F   21   7D   40   ED   5B
0C   40   FD   21   00   00   CD   A9
40   2A   10   40   ED   5B   14   40
1B   CD   A9   40   FD   E5   FD   21
00   40   CD   2B   0F   C1   C9   E5
A7   ED   52   E1   C8   4E   06   00
FD   09   23   18   F2   21   27   00
ED   5B   04   40   19   FD   21   90
40   FD   75   01   FD   74   02   FD
75   0C   FD   74   0D   FD   21   00
40   21   82   40   01   35   00   ED
B0   ED   4B   04   40   C9   S    →
```

Em caso de erro, recomece a segunda etapa.

3. Tecle **POKE 16510,0. NEW LINE.** Seu programa aparecerá como a listagem 3

```
                    LISTAGEM - 3

   0 REM LN 7?5?RND GOSUB ?£RND
CLEAR 5  LN ▩RNDE(RND GOSUB ?=RN
D.LN ▩RND CLEAR FAST CLEAR 5 RND
LN F?AT TAN FAST ▩ GOSUB ? LPRIN
T COS ?▘ CLEAR ▬7/ PAUSE 5B  GO
SUB ?▪RND; CLEAR 5▩RND CLEAR ?▘
CLEAR ?▘ CLEAR ?£ CLEAR ?$ CLEAR
 5 RND5▙RND▘P  GOSUB ▩ GOSUB ?▪R
NDTAN
  20 LET X$=""
  30 LET X=16514
  40 IF X$="" THEN INPUT X$
  50 IF X$(1)="S" THEN STOP
  60 PRINT X$(1 TO 2);"  ";
  70 POKE X,(16*CODE X$+CODE X$(
2)-476)
  80 LET X=X+1
  90 LET X$=X$(3 TO )
 100 GOTO 40
```

Apague as linhas **20** a **100** e tecle:

```
                    LISTAGEM - 4

  10 SAVE "SOMA"
  20 PRINT "SOMA SINTATICA=";USR
16514
  30 PRINT "CONFERE? S/N"
  40 IF INKEY$="S" THEN GOTO 60
  50 GOTO 40
  60 PRINT "SOMA SINTATICA NO EN
DERECO ";
  70 INPUT RTP
  80 POKE 16389,INT (RTP/256)
  90 POKE 16388,RTP-INT (RTP/256
)*256
 100 PRINT USR 16567
 110 IF INKEY$="" THEN GOTO 110
 120 NEW
```

4. Salve a versão final da **Soma Sintática,** teclando RUN. Assim ela estará pronta para ser usada.

A versão gravada na fita, ao ser lida corretamente, deverá mostrar a seguinte mensagem:

```
SOMA SINTATICA = 25525
```

Para usar esta mensagem, tecle o endereço para o qual você quer baixar o RTP — Ex.: "Se você tem 16K, 30000" — o programa responderá com o mesmo valor. A **Soma Sintática** será então obtida com:

PRINT USR (endereço que você escolheu).

Lembre-se agora de **carregar** a fita "SOMA" antes de teclar ou **carregar** seu programa. Após informar o endereço limite da memória utilizada, **tecle** qualquer tecla para limpar a área do BASIC.

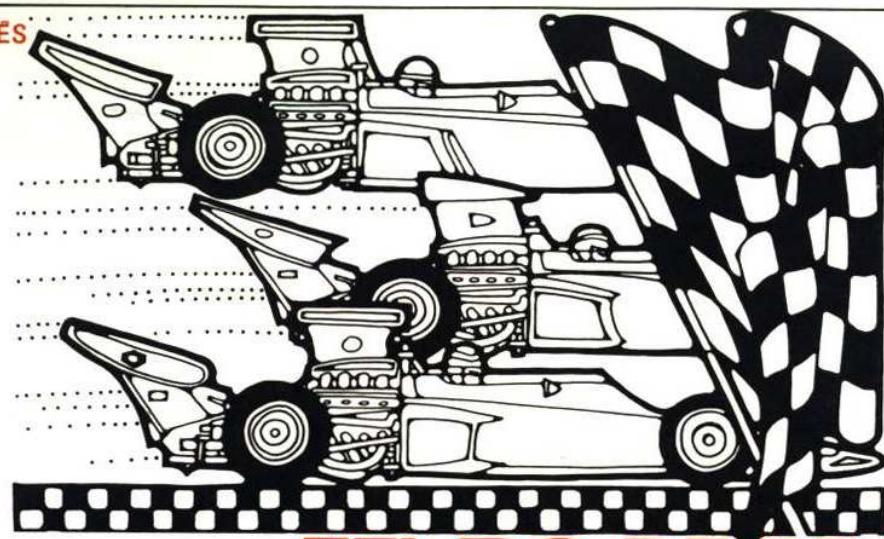
Convém observar que a **Soma Sintática** checa o programa e as variáveis — caso elas existam — e portanto deve ser usada imediatamente após a carga do seu programa.

Em programas que iniciam rodando após a carga, é conveniente proceder como está especificado abaixo; onde XXX, YYY e ZZZ são linhas do seu programa:

1 PRINT "SOMA SINTÁTICA = ";
2 PRINT USR (endereço escolhido)
3 PRINT "CONFIRA E TECLE NEW LINE PARA CONTINUAR"
4 PAUSE 6E4 (6 x 10⁴)

Seu Programa {

XXX STOP
YYY SAVE "Nome do seu programa"
ZZZ GOTO 1

Ao salvar seu programa, anote a **Soma Sintática** junto ao nome do programa, no rótulo da fita. Caso o programa, ao ser carregado, não dê o mesmo valor, carregue-o novamente.

TEMPO DE REAÇÃO

Carlos Eduardo R. Salvato

Uma das consequências da ingestão de álcool pelo ser humano é a perda do senso de auto-crítica: por isso raramente um bêbado admite que está bêbado.

Imagine a seguinte situação: Você como anfitrião de uma festinha de sábado à noite. Um de seus convidados, visivelmente "alto", pega as chaves do carro e resolve voltar para casa. É óbvio que você tenta demovê-lo da idéia, mas ele, como está bêbado, não tem consciência do perigo e teima em sair. Qual a solução? pegá-lo pela curiosidade e desafio frente a um computador. Uma sugestão: Desafiá-lo a medir a rapidez de seus reflexos em um TK. Logo o desafio vai se generalizar e todos os seus convidados estarão competindo para quebrar o recorde (Aqui na redação, a Nancy é a nossa recordista e já foi aclamada como um exemplo da óbvia superioridade feminina). O desafio proposto a todos irá permitir que você sirva um café sem açúcar ao seu amigo e aos demais convidados, dissipando assim, os "vapores etílicos", sempre presentes em reuniões de sábado à noite".

Ao iniciar o programa, o computador coloca as instruções na tela. Digitando "S", tente começar.

```
ESSE PROGRAMA VAI MOSTRAR OS RE-
SULTADOS DE 10 TESTES DE REACAO.


QUANDO "S" FOR PRESSIONADO UM
NUMERO ALEATORIO IRA' APARECER NO
QUADRADO ABAIXO.ENTRE COM O NU-
MERO O MAIS RAPIDAMENTE POSSIVEL
ISSO ACONTECERA' ATE' O DECIMO
TESTE SER COMPLETADO.
APOS O GRAFICO,DIGITE NEW LINE.

        NUMERO :-
```

Figura 1

No quadrado, aparecerá um algarismo de Ø a 9 e você deve digitá-lo o mais rapidamente possível. Acertando, ele desaparece e outro algarismo aparece no quadrado. Ao terminar o teste, o computador apresenta os resultados abaixo:

Figura 2

Se você obteve o tempo mais rápido (ou se você foi o primeiro), seu recorde ficará registrado por toda a eternidade. . . (ou seja, até o computador ser desligado). Você poderá registrar o recorde, gravando outra vez o programa em outra fita, usando o seguinte comando:

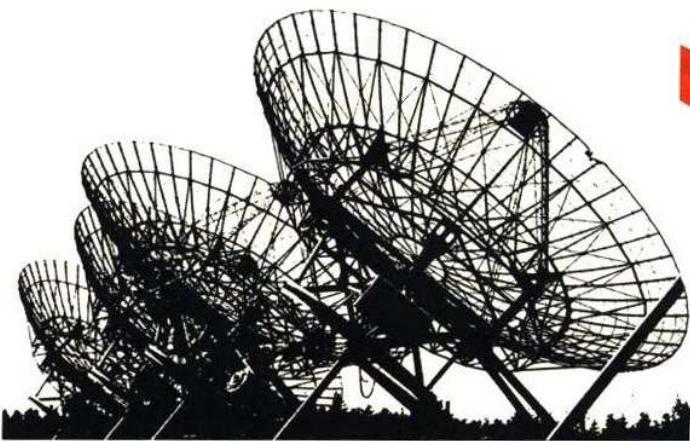
GOTO 1Ø7Ø

Após digitar o programa da listagem anexa, jogue pelo menos uma vez, iniciando com um RUN. A seguir, grave o programa em fita usando o seguinte comando:

GOTO 1Ø7Ø

Este programa pode ser rodado num TK82 ou TK83 com expansão; num TK85 ou similar.

```
  10 REM MICROMEGA - 1983
  20 SLOW
  30 LET F=100000
  40 DIM N(10)
  50 CLS
  60 RAND
  70 LET Q=100000
  80 LET S=0
  90 LET R=0
 100 PRINT AT 4,0;"ESSE PROGRAMA
 VAI MOSTRAR OS RE-"
 110 PRINT AT 5,0;"SULTADOS DE 1
0 TESTES DE REACAO."
 120 PRINT AT 9,0;"QUANDO ""S""
FOR PRESSIONADO UM"
 130 PRINT "NUMERO ALEATORIO IRA
▪APARECER NO"
 140 PRINT "QUADRADO ABAIXO.ENTR
E COM O NU-"
 150 PRINT "MERO O MAIS RAPIDAME
NTE POSSIVEL"
 160 PRINT "ISSO ACONTECERA▪ ATE
▪ O DECIMO  TESTE SER COMPLETADO
."
 170 PRINT "APOS O GRAFICO,DIGIT
E NEW LINE."
 180 PRINT AT 16,16;"▒▒▒▒▒"
 190 PRINT AT 17,16;"▒   ▒"
 200 PRINT AT 18,16;"▒   ▒"
 210 PRINT AT 19,16;"▒   ▒"
 220 PRINT AT 20,16;"▒▒▒▒▒"
 230 PRINT AT 18,5;"NUMERO  :  "
 240 POKE 16437,255
 250 PAUSE 30000
 260 IF INKEY$="S" THEN GOTO 280

 270 GOTO 240
 280 FOR K=1 TO 10
 290 PRINT AT 18,18;" "
 300 FOR O=0 TO INT (RND*60)
 310 NEXT O
 320 LET A=INT (RND*10)
 330 PRINT AT 18,18;A
 340 FOR T=0 TO 100000
 350 POKE 16437,255
 360 IF CODE INKEY$=A+28 THEN GO
TO 380
 370 NEXT T
 380 LET N(K)=T
 390 NEXT K
 400 CLS
 410 PRINT AT 1,1;"TESTE DE REAC
AO"
 420 PRINT AT 2,1;"=============
=="
 430 PRINT AT 4,1;"RESULTADOS :-
"
 440 PRINT AT 5,1;"-----------"
 450 PRINT AT 7,0;"+ RAPIDO  :"
 460 PRINT AT 9,0;"+ DEVAGAR  :"
 470 PRINT AT 12,0;"MEDIA  :-"
 480 PRINT AT 21,22;"..TESTES.."
 490 PRINT AT 5,19;"T";AT 6,19;"
E";AT 7,19;"M";AT 8,19;"P"
 500 PRINT AT 14,19;"S";AT 15,19
;"E";AT 16,19;"G";AT 17,19;"S"
 510 FOR O=2 TO 20
 520 PRINT AT O,21;"-"
 530 NEXT O
 540 LET A=0
 550 FOR K=1 TO 10
 560 LET N(K)=(INT ((N(K)/.02)+.
5))/1000
 570 IF N(K)<Q THEN LET Q=N(K)
 580 IF N(K)>S THEN LET S=N(K)
 590 LET A=A+N(K)
 600 NEXT K
 610 LET A=(INT ((A*100)+.5))/10
00
 620 PRINT AT 7,11;Q
 630 PRINT AT 9,11;S
 640 PRINT AT 12,11;A
 650 LET B=(INT ((Q*.9)*10))/10
 660 LET C=(INT ((S*1.1)*10))/10
 670 PRINT AT 20,17:B
 680 PRINT AT 2,17;C
 690 PRINT AT 11,17;(B+C)/2
 700 LET D=(C-B)/36
 710 FOR Q=1 TO 10
 720 LET E=(N(Q)-B)/D
 730 FOR Z=2 TO E+1
 740 PLOT 43+(2*Q),Z
 750 NEXT Z
 760 NEXT Q
 770 IF A<F THEN GOSUB 850
 780 IF A<F THEN LET F=A
 790 PRINT AT 16,0;"MEDIA + RAPI
DA"
 800 PRINT AT 18,5;F
 810 PRINT AT 20,1;"POR ";A$
 820 IF INKEY$<>"" THEN GOTO 40

 830 GOTO 820
 840 LET R=0
 850 LET B$="▒ VOCE OBTEVE ▒"
 860 LET C$="MEDIA + RAPIDA"
 870 LET D$="ENTRE COM NOME"
 880 LET E$="                "
 890 PRINT AT 16,0;E$
 900 PRINT AT 20,0;E$
 910 FOR O=0 TO 20
 920 NEXT O
 930 LET R=R+1
 940 PRINT AT 18,0;C$
 950 PRINT AT 16,0;B$
 960 PRINT AT 20,0;D$
 970 FOR O=0 TO 20
 980 NEXT O
 990 IF R<3 THEN GOTO 890
1000 INPUT A$
1010 IF LEN A$>10 THEN GOTO 1000
1020 FOR O=16 TO 20 STEP 2
1030 PRINT AT O,0;E$
1040 NEXT O
1050 RETURN
1060 STOP
1070 SAVE "REACA▒"
1080 GOTO 40
1090 STOP
```

MENSA International

L.G.M.: MENSAGEM DE VEGA ?

"Era uma vez um macaco inteligente,
Que vivia a soprar fagote
Porque, dizia ele, provavelmente,
Dentro de bilhões de anos
Eu certamente conseguirei
Tocar alguma música".

Sir Arthur Eddington

A idéia de contactar civilizações de outros "mundos" não é recente entre os homens. Deixando de lado as crendices populares e as mediocridades publicitárias, a imaginação humana chegou a produzir muitas obras sobre esse assunto. Clássicos literários como **"Micrômegas"**, de Voltaire; "História Cômica dos Estados e Impérios da Lua e do Sol", de Cyrano de Bergerac e **"Da Terra à Lua"** de Júlio Verne são, juntamente com um grande número de **"Science Fiction"**, bons testemunhos de boa qualidade (Figura 1).

Figura 1

Renato da Silva Oliveira

Também nos meios de comunicação de massa, são inúmeros os exemplos da procura de outros mundos, como a novela radiofônica "Guerra dos Mundos" narrada por **Orson Welles**, no início do século. Esta novela teve maior repercussão que os "contos de fadas" cinematográficos **"Close Encounters"** e **"E.T."** de Steven Spielberg que, por assim dizer, invadiram facilmente a Terra aproveitando-se da boa vontade das pessoas.

Até mesmo o "Gran Circo EEUU" produziu, bem ao seu estilo, um famoso relatório sobre possíveis seres extraterrestres no famoso "Livro Azul" da U.S. Air Force.

Como não poderia deixar de ser, os cientistas produziram, também, alguns trabalhos (provavelmente os mais sérios e audaciosos), sobre sua velha idéia, assim que surgiram oportunidades propícias. Desde grandes equívocos, como os "Canais Artificiais de Marte" e os "Little Green Men (L.G.M.)" até as mensagens levadas pelas sondas PIONER X e XI e VOYAGER I e II.

Figura 2

Esta seção conta com a colaboração dos membros da **MENSA INTERNACIONAL** *— brasileiros e estrangeiros residentes no Brasil.*

A **MENSA** *é uma organização sem fins lucrativos, originária da Inglaterra e que conta com milhares de membros espalhados pelo mundo inteiro. Entre estes, podemos citar* **Isaac Asimov, Arthur C. Clarke, Clive Sinclair** *e o recentemente falecido* **R. Buckminister Fuller** *(o poeta que inventou a* **Cúpula Geodésica**).

Os principais objetivos e bases de atuação da MENSA são: promover e encorajar o desenvolvimento da inteligência humana para benefício da humanidade, incentivar a pesquisa sobre a natureza e promover um ambiente socialmente estimulante para seus sócios.

Para o ingresso na MENSA é necessário apenas que o candidato se submeta a um teste homologado pela MENSA INTERNATIONAL, superando o nível médio de inteligência de 98 por cento da humanidade.

Informações a respeito, podem ser obtidas através de carta enviada para esta seção, no seguinte endereço:

MICROHOBBY
SEÇÃO QUEBRA-CABEÇA
Rua Bahia, 1049
Cep 01244 — São Paulo

Provavelmente, um dos mais interessantes empreendimentos científicos visando a comunicação com possíveis sêres extraterrestres, foi a mensagem enviada ao espaço com o radiotelescópio de Arecibo em Puerto Rico (o maior radiotelescópio de espelho esférico do mundo, com cerca de 305 metros de diâmetro).

Figura 3

O interesse dos cientistas em contactar com outras (possíveis) civilizações não é gratuito.

Apesar de altamente improvável, a pequena chance de sucesso justifica o esforço desprendido. Atualmente, através de uma equação simplificada, obtida pelo radioastrônomo Frank Drake na década de 60, estima-se que dentre aproximadamente 200 bilhões de estrelas de nossa galáxia, cerca de um milhão (menos de 0,001 por cento) têm condições de abrigar civilizações tecnologicamente desenvolvidas. Mesmo sendo otimista e supondo-se que todas elas sejam "povoadas", a probabilidade de conseguirmos nos comunicar é ainda desprezivelmente pequena.

$$N = R^{*} \cdot fg \cdot fp \cdot ne \cdot fi \cdot ft \cdot L$$

Equação de Frank Drake

onde:

- **N = número de civilizações tecnológicas da Galaxia**
- **R^{*} = taxa média de formações de estrelas da Galaxia**
- **fg = fração de estrelas com condições de possuir sistemas planetárias**
- **fp = fração de estrelas que possuem um sistema planetário**
- **ne = número de planetas habitáveis por sistema planetário**
- **fe = probabilidade de surgimento de vida num planeta**
- **fi = fração de planetas habitáveis com vida inteligente**
- **ft = probabilidade de uma civilização desenvolver tecnologia**
- **L = longevidade de uma civilização**

Figura 4

De certo modo, estamos fazendo — com algumas desvantagens —, a vez do macaco dos versos de Sir Arthur Eddington, transcrito no início deste artigo. O macaco pelo menos sabia que tocar música é possível, enquanto nós (humanos), não sabemos sequer se existem outras civilizações além da nossa.

Se você teve paciência de ler tudo até esta linha, ao menos justifique seu trabalho lendo ainda as que se seguem. Você verá que tudo se relaciona ao tema de nosso **"Quebra-Cabeça"** deste mês: **"L.G.M. ou Mensagem de Vega?".**

Citamos, de passagem, o equívoco científico dos "L.G.M.(s)". Essa sigla surgiu para os quatro primeiros pulsares descobertos: L.G.M.; LGM_2, LGM_3 e LGM_4. O primeiro deles foi descoberto em meados de 1967 por uma jovem radioastrônoma, Jocellyn Bell, através do radiotelescópio da Cambridge University. Esses "objetos" são notados, devido as intensas e regularíssimas emissões de micro-ondas que deles nos chegam (em geral cerca de uma por segundo). A esperança quase religiosa dos astrônomos em detectar vida fora da Terra, fez com que batizassem — oficiosamente —, as quatro primeiras dessas fontes, de ' Little Green Men", pensando tratar-se de "sinais inteligentes". Logo, porém, muitos outros pulsares foram descobertos e o equívoco "L.G.M." foi desfeito. Hoje, sabemos que os pulsares são provavelmente, estrelas neutrônicas em alta rotação.

Essa estória ilustra, muito bem, a boa vontade e o entusiasmo dos cientistas em admitirem como possível a comunicação com civilizações extra-terrestres (se elas existirem, é claro!).

continua

MICROHOBBY

L.G.M.

O *Quebra-Cabeça* deste mês surgiu de uma estória semelhante:

"Um dos nossos leitores, o Sr. Nabor Rosenthall, construiu em seu sítio próximo à pequena cidade de São Tomé, um radiotelescópio "caseiro", a qual tem, acoplado, um TK85. No primeiro dia de setembro deste ano, precisamente entre as 21 e 23 horas, ele recebeu uma série de pulsos de micro-ondas, provenientes de um ponto do céu cujas coordenadas são:

Declinação: δ = N 38º 46'
Ascenção Reta: α = 18 h 36' 8''

Essas são, exatamente, as coordenadas da estrela α *Lyrae* (Vega).

Com a inteligência e perspicácia características de nossos leitores, o Sr. Nabor percebeu que, se associasse aos intervalos — durante os quais recebia os pulsos —, o valor 1 (um) e aos períodos "mudos" entre os pulsos, associasse o valor Ø (zero), obteria uma seqüência de *uns* e *zeros*, aparentemente regulares que se repetiria, mais ou menos a cada uma hora.

Assim sendo, o Sr. Nabor, por não ter ainda conclusões convincentes, resolveu nos escrever para que pudéssemos auxiliá-lo a chegar a tal conclusão e também solicitou-nos que tentássemos obter informações com outros leitores, a respeito da seqüência: *"se ela é ou não uma mensagem e em caso positivo, qual o seu significado?"*.

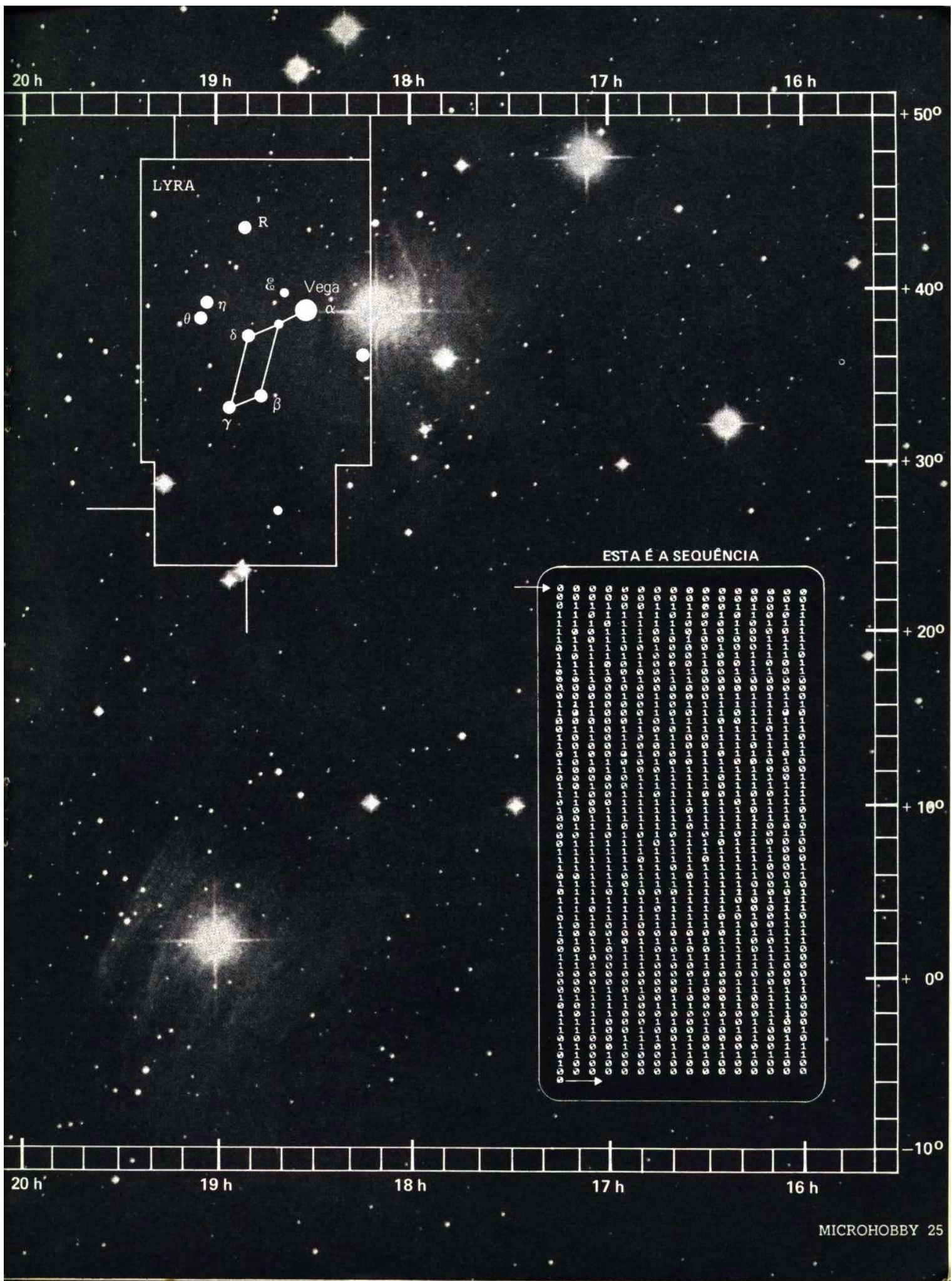
A seqüência está listada a seguir e o *Quebra-Cabeça* consiste em — com o auxílio do TK —, descobrir se ela é um novo equívoco do tipo *"L.G.M."* ou se é realmente uma mensagem "E.T.", proveniente de Vega. Se a resposta for positiva, vocês devem decodificá-la e interpretar o seu significado.

A ordem em que os pulsos (Øs e 1's) foram recebidos são indicados pelas setas no início e no fim da listagem.

20 h
19 h
18 h
17 h
16 h
+ 50º
+ 40º
+ 30º
+ 20º
+ 10º
+ 0º
–10º
LYRA
R
Vega
ESTA É A SEQUÊNCIA

DICAS

AS FUNÇÕES "D" DO TK85

Nancy Mitie Ariga

Aqui está, num artigo escrito por nossa analista de Software, a resposta da carta escrita por nosso leitor **Fernando José Strobel,** que pedia informações a respeito de **gravação de fita de dados.** Um pedaço do texto da carta era o seguinte: "Peço que a sua revista publique um artigo, explicando a armazenagem de dados em fita, como um exemplo bem complexo, com **fichas lógicas** e **fichas físicas".**

Para resolvermos sua dúvida a respeito das funções de armazenamento para o TK85, fizemos um programa utilizando os comandos DLOAD, DSAVE, DVERIFY e o mesmo pode ser feito para os comandos DHLOAD e DHSAVE, mas em **Hi-Speed.**

Através desses comandos, é possível o armazenamento e a recuperação de informações numa fita cassete, permitindo assim, o uso de um grande arquivo de dados. Esses comandos são utilizados com a função USR, acompanhada do endereço inicial da subrotina, em linguagem de máquina do comando a ser utilizado.

COMANDO	ENDEREÇO INICIAL	FUNÇÃO DE CADA COMANDO
DSAVE	8288	Gravação do arquivo
DVERIFY	9816	Verifica a gravação
DLOAD	83Ø5	Lê a fita e armazena no **buffer** indicado
DHSAVE	9ØØ8	Grava na fita em **Hi-Speed**
DHLOAD	9189	Lê a fita em **Hi-Speed**

Para melhor compreensão, dividimos o programa em várias partes, portanto, digite todas as linhas exatamente como estão, porque elas formarão, no final, o seu programa mestre.

O programa começa armazenando os endereços de cada subrotina, em linguagem de cada comando:

```
PROGRAMA 1
4 LET DSAVE=8288
5 LET DVERIFY=9816
6 LET DLOAD=8305
7 LET DHSAVE=9008
8 LET DHLOAD=9189
```

Figura 1

Vamos inicialmente criar uma **ficha física** com 10 **fichas lógicas.** Mas, o que é uma **ficha física?** E uma **ficha lógica?**

Suponhamos que você deseje um arquivo com nomes e telefones. Um determinado nome com o seu respectivo telefone, formam uma **ficha lógica.** Um grupo de **fichas lógicas** (no nosso caso 10) formam uma **ficha física.**

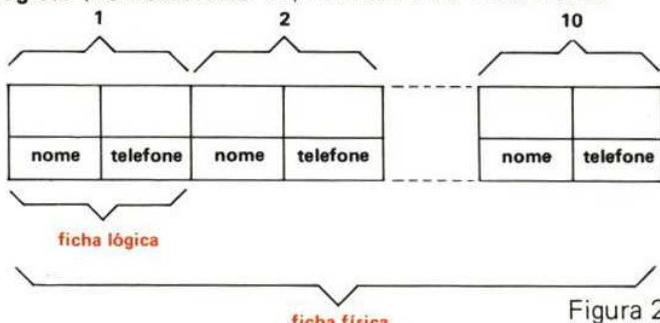



Figura 2

Digitando o programa **2,** você criará uma **ficha física** que, posteriormente será gravada. Cada **ficha lógica,** conterá um nome com no máximo 15 dígitos e um telefone com no máximo 8 dígitos que serão armazenados no **buffer,** formando a **ficha física** (B$) que poderá conter no máximo, 230 bytes.

1 ficha lógica → 15 bytes do nome + 8 bytes do telefone = 23 bytes.

10 fichas lógicas → 230 bytes

```
PROGRAMA 2
1000 REM CRIACAO DE FICHA FISICA
1005 CLS
1010 DIM B$(230)
1020 DIM N$(15)
1030 DIM T$(8)
1040 LET A=1
1050 FOR I=1 TO 10
1060 SCROLL
1070 PRINT "NOME=";
1080 INPUT N$
1090 PRINT N$
1100 SCROLL
1110 PRINT "TELEF.=";
1120 INPUT T$
1130 PRINT T$
1140 SCROLL
1150 PRINT
1160 LET C$=N$+T$
1170 LET B$(A TO A+LEN C$-1)=C$
1180 LET A=A+LEN C$
1190 NEXT I
1200 SCROLL
1210 PRINT "FICHA FISICA COMPLET
A"
1213 SCROLL
1214 PRINT "P/GRAVAR DIGITE (NEW
 LINE)"
1220 IF INKEY$="" THEN GOTO 1220
```

Figura 3

Digite RUN e aparecerá na tela:

```
NOME=

"■"
```

Figura 4

Digite um nome e NEW LINE:

```
NOME=MICROHOBBY
TELEF.=

"■"
```

Figura 5

Digite um telefone e NEW LINE:

```
NOME=MICROHOBBY
TELEF.=256.8348

NOME=

"■"
```

Figura 6

e assim por diante.

Agora que você possui uma **ficha física** em seu **buffer**, vamos gravá-lo. Digite o programa **3**, que possui o comando DSAVE.

```
PROGRAMA 3

2000 REM GRAVACAO FICHA FISICA
2005 CLS
2006 PRINT "P/GRAVAR APERTE RECO
RD DO GRAVA-DOR E NEW LINE"
2007 IF INKEY$="" THEN GOTO 2007
2010 LET Z=LEN B$
2020 LET Z$="B"
2030 LET BC=USR DSAVE
2040 SLOW
2050 IF BC=0 THEN PRINT AT 7,3;"
OK"
2060 IF BC<>0 THEN PRINT AT 7,3;
"ERRO=";BC
2070 PRINT "NOVA GRAVACAO (S/N)?
"
2080 IF INKEY$="S" THEN GOTO 200
0
2085 IF INKEY$="N" THEN GOTO 300
0
2090 GOTO 2080
```

Figura 7

Em **Z** está definido o número de **bytes** do **buffer** escolhido e que será gravado. Z$ seleciona o **buffer** que será gravado (no nosso exemplo Z$ = "B", porque em B$ estão armazenados os dados da **ficha física**).

BC é uma variável qualquer, criada para utilizarmos a função USR, onde após a execução do comando DSAVE, será armazenado o código de reportagem. (Ver código de reportagem, página 29.7 do manual do TK85).

Certifique-se de que o MIC do TK está conectado com o MIC do seu gravador, aperte o RECORD e digite NEW LINE. O comando DSAVE será ativado e a primeira **ficha física** estará sendo gravada.

Se a subrotina DSAVE foi realizada corretamente (BC = Ø), aparecerá na tela um "OK". Caso contrário, o código de reportagem aparecerá na tela (consulte o manual do TK85, página 29.7).

IMPORTANTE: Após o término de cada comando de armazenamento, devemos retornar ao modo SLOW, porque o computador entra no modo FAST com o início da subrotina, em linguagem de máquina.

Agora que você gravou em sua fita cassete, certifique-se que sua gravação foi feita corretamente, digitando o programa 4.

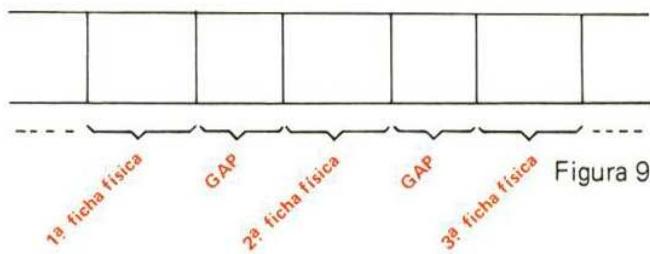
```
PROGRAMA 4

3000 REM VERIFICACAO DA GRAVACAO
3005 CLS
3010 PRINT "P/VERIFICACAO REBOBI
NE A FITA"
3020 PRINT "APERTE O PLAY DO GRA
VADOR"
3030 PRINT "DIGITE NEW LINE"
3040 IF INKEY$="" THEN GOTO 3040
3050 LET BC=USR DVERIFY
3060 SLOW
3063 PRINT
3065 IF BC=0 THEN PRINT "OK"
3070 IF BC<>0 THEN PRINT "ERRO="
;BC
3080 PRINT "NOVA VERIFICACAO (S/
N)?"
3085 IF INKEY$="S" THEN GOTO 300
0
3088 IF INKEY$="N" THEN GOTO MEN
U
3100 GOTO 3085
```

Figura 8

Conecte o EAR do TK85 com o EAR do seu gravador. Diminua um pouco o volume do gravador, aperte o PLAY e digite NEW LINE. O comando DVERIFY foi ativado, atuando como se estivesse lendo a fita, mas não armazena os dados na RAM, permitindo a você, verificar se as informações foram gravadas corretamente sem destruir o conteúdo da RAM. Após a execução da subrotina aparecerá na tela, o código de reportagem. Se aparecer o código de erro diminua, pouco a pouco, o volume do gravador até surgir um "OK", significando que a gravação foi boa e que a fita será lida neste volume de som; caso contrário, será informado o erro.

Deseja outras **fichas físicas** em sua fita cassete? Ela pode conter inúmeras **fichas físicas,** que devem ser gravadas uma após a outra. Sua fita ficará assim:

Figura 9

continua

GAP é um espaço entre suas **fichas físicas** de aproximadamente cinco segundos ou mais, antes que a segunda **ficha física** seja lida.

Vamos utilizar as informações armazenadas em sua fita cassete, através de uma pesquisa. Digite o programa **5**.

```
PROGRAMA 5

4000 REM PESQUISA
4005 CLS
4010 PRINT AT 7,3;"VOCE DESEJA:"
4020 PRINT AT 9,3;"1. NOME "
4030 PRINT AT 11,3;"2. TELEFONE
"
4040 PRINT AT 13,3;"3. RELACAO D
OS NOMES COMECA-        DOS COM D
ETERMINADA LETRA"
4050 IF INKEY$="1" THEN GOTO 500
0
4060 IF INKEY$="2" THEN GOTO 600
0
4070 IF INKEY$="3" THEN GOTO 700
0
4080 GOTO 4050
5000 REM PESQUISA DE NOME
5005 CLS
5010 PRINT "QUAL O TELEFONE ?"
5020 INPUT T$
5030 PRINT T$
5035 PRINT
5040 PRINT "REBOBINE A FITA"
5050 PRINT "APERTE O PLAY DO GRA
VADOR"
5060 PRINT "DIGITE NEW LINE"
5065 IF INKEY$="" THEN GOTO 5065
5070 LET Z=0
5075 DIM B$(230)
5080 LET Z$="B"
5090 LET BC=USR DLOAD
5095 IF BC=1 THEN GOTO 5555
5100 IF BC<>0 THEN GOTO 5500
5120 FOR I=1 TO 230 STEP 23
5130 IF B$(I+15 TO I+22)=T$ THEN
 GOTO 5600
5140 NEXT I
5300 GOTO 5070
5500 SLOW
5510 PRINT "ERRO=";BC
5515 PRINT
5520 PRINT "APERTE STOP DO GRAVA
DOR"
5530 PRINT "APERTE NEW LINE P/ N
OVA PESQUISA"
5540 IF INKEY$="" THEN GOTO 5540
5550 GOTO 4000
5555 SLOW
5656 PRINT
5557 PRINT
5558 PRINT
5559 PRINT "NAO FOI ACHADO ESTE
TELEFONE       NA FITA"
5560 PRINT
5562 PRINT "NOVA PESQUISA (S/N)?
"
5564 IF INKEY$="S" THEN GOTO 400
0
5565 IF INKEY$="N" THEN GOTO MEN
U
5566 GOTO 5564
5600 SLOW
5605 PRINT
5610 PRINT "APERTE STOP DO GRAVA
DOR"
5615 PRINT
5620 PRINT "NOME=";B$(I TO I+14)
5630 PRINT "TELEFONE=";B$(I+15 T
O I+22)
5640 PRINT
5645 PRINT "VOCE DESEJA NOVA PES
QUISA ? "
5646 PRINT "             (S/N)"
5650 IF INKEY$="S" THEN GOTO 400
0
5660 IF INKEY$="N" THEN GOTO MEN
U
5670 GOTO 5650
6000 REM PESQUISA DE TELEFONE
6005 CLS
6010 PRINT "QUAL O NOME ?"
6015 DIM N$(15)
6020 INPUT N$
6030 PRINT N$
6035 PRINT
6040 PRINT "REBOBINE A FITA"
6050 PRINT "APERTE O PLAY DO GRA
VADOR"
6060 PRINT "DIGITE NEW LINE"
6070 IF INKEY$="" THEN GOTO 6070
6080 LET Z=0
6085 DIM B$(230)
6090 LET Z$="B"
6100 LET BC=USR DLOAD
6103 IF BC=1 THEN GOTO 9500
6110 IF BC<>0 THEN GOTO 5500
6115 FOR I=1 TO 230 STEP 23
6120 IF B$(I TO I+14)=N$ THEN GO
TO 5600
6130 NEXT I
6140 GOTO 6080
6200 SLOW
6210 PRINT
6220 PRINT
6230 PRINT
6240 PRINT "NAO FOI ACHADO ESTE
NOME NA FITA"
6300 PRINT
6310 PRINT "NOVA PESQUISA (S/N)?
"
6312 IF INKEY$="S" THEN GOTO 400
0
6314 IF INKEY$="N" THEN GOTO MEN
U
6330 GOTO 6312
7000 REM PESQUISA POR LETRA
7005 LET N=1
7007 DIM N$(1000)
7010 CLS
7020 PRINT "NOMES COMECADOS COM
A LETRA:";
7030 INPUT L$
7040 PRINT L$
7045 PRINT
7050 PRINT "REBOBINE A FITA"
7060 PRINT "APERTE O PLAY DO GRA
VADOR"
7070 PRINT "DIGITE NEW LINE"
7080 IF INKEY$="" THEN GOTO 7080
7090 LET Z=0
7095 DIM B$(230)
7100 LET Z$="B"
7110 LET BC=USR DLOAD
7115 IF BC=1 THEN GOTO 8500
7120 IF BC<>0 THEN GOTO 5500
7130 FOR I=1 TO 230 STEP 23
7140 IF B$(I)=L$ THEN GOSUB 8000
7150 NEXT I
7160 GOTO 7090
7200 SLOW
7210 PRINT "ERRO=";BC
7220 PRINT
7230 PRINT "APERTE O STOP DO GRA
VADOR"
7240 PRINT "DIGITE NEW LINE P/NO
VA PESQUISA"
7245 GOTO 4000
8000 LET N$(N TO N+22)=B$(I TO I
+22)
8100 LET N=N+23
8200 RETURN
8500 CLS
8503 SLOW
8505 IF N=1 THEN GOTO 9600
8509 FOR U=1 TO N+22 STEP 23
8510 PRINT N$(U TO U+22)
8520 NEXT U
8620 GOTO 5640
```

```
9600 PRINT
9602 PRINT
9603 PRINT
9605 PRINT  "NAO FOI ENCONTRADO N
ENHUM NOME  COMECADO COM ESTA LE
TRA"
9610 GOTO 5645
```

Figura 10

A variável **Z** foi criada e zerada. Nela será armazenado o número de bytes lidos, que serão armazenados no **buffer**, após a execução do comando DLOAD.Z$ indica o **buffer** em que serão armazenadas as informações lidas na fita.

Ao ativarmos o comando **DLOAD**, o computador coloca no **buffer** a primeira **ficha física** e por cinco segundos podemos manusear os dados antes que a outra **ficha** entre no **buffer.**

O programa **5** possui três tipos de pesquisa:
1. Procura o nome cujo telefone você forneceu em T$.
2. Procura o telefone de um determinado nome, indicado em N$.
3. Lista os nomes começados com uma determinada letra, indicada em L$.

Mas você pode criar outros tipos de pesquisa. Procure entender como o comando DLOAD funciona e como é feito a pesquisa; depois, deixe livre sua imaginação!

Cada **ficha lógica** contém 23 bytes, sendo os 15 primeiros bytes, reservados para o nome e os outros 8 bytes, para o telefone.

Assim, se você deseja saber o telefone de "X", o computador separa as **fichas lógicas** (23 em 23 bytes) e compara os 15 primeiros bytes com o nome que está em N$. Se forem iguais, aparece na tela o nome e o respectivo telefone; caso contrário, analisa a próxima **ficha lógica.** Se este nome não for encontrado nesta **ficha física,** inicia a leitura da próxima **ficha física** e o mesmo processo se repete.

Caso chegue ao fim do arquivo de dados sem achar o nome, aparecerá na tela, o seguinte:

```
NAO FOI ACHADO ESTE NOME NA FITA
```

Figura 11

O mesmo processo é feito para a pesquisa do nome cujo telefone você informou em T$. A única diferença é que são comparados os oito últimos bytes de cada **ficha lógica,** com o telefone em T$.

Na pesquisa de nomes começados com uma letra (L$) é comparado o primeiro byte de cada **ficha lógica** com L$. Se forem iguais, esta **ficha lógica** é armazenada em um outro **buffer** N$; caso contrário, analisa a próxima **ficha lógica.**

Quando o código de reportagem em BC acusar "fim de arquivo", as informações que estão no **buffer** N$ são imprimidos na tela.

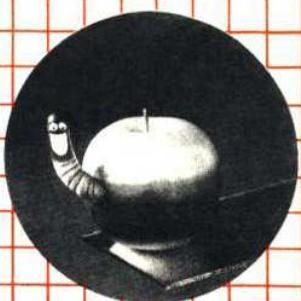
Para completar o seu programa, digite o programa **seis** que é o MENU.

```
PROGRAMA 6

 10 REM MENU
 12 LET MENU=10
 13 CLS
 15 PRINT AT 6,8;"M E N U"
 20 PRINT AT 8,5;"VOCE DESEJA:"
 30 PRINT AT 10,5;"1. CRIAR FIC
HA FISICA"
 40 PRINT AT 12,5;"2. GRAVAR FI
CHA FISICA"
 50 PRINT AT 14,5;"3. VERIFICAR
 GRAVACAO"
 60 PRINT AT 16,5;"4. PESQUISAR
 FICHA LOGICA"
 70 IF INKEY$="1" THEN GOTO 100
0
 80 IF INKEY$="2" THEN GOTO 200
0
 90 IF INKEY$="3" THEN GOTO 300
0
 100 IF INKEY$="4" THEN GOTO 400
0
 110 GOTO 70
```

Figura 12

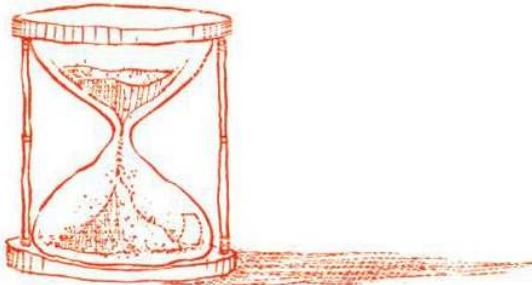
Agora que você sabe como os comandos de armazenamento e recuperação de dados funcionam, crie outras fichas lógicas e outros programas-pesquisa.

por dentro do apple

® Apple é marca registrada de Apple Computer, Inc.

Autor: Prof. Wilson José Tucci – Coordenador de Projetos Especiais da Escola Experimental Pueri Domus.

A AMPULHETA ELETRÔNICA

Com este programa abrimos a seção RELÓGIOS, onde pretendemos apresentar alguns programas para marcar o tempo no seu Apple; nada melhor do que começar com um relógio bem primitivo: a ampulheta (programas simulando relógios mais antigos, como o relógio de sol, serão bem aceitos pela nossa redação). Com a colaboração dos leitores iremos evoluindo até os modernos relógios digital-analógicos.

Se você nunca viu uma ampulheta, nem mesmo na abertura do "Túnel do Tempo", eis como ela se parece:

O tempo que a areia leva para passar da parte de cima para a parte de baixo, completa um ciclo.

As primeiras ampulhetas, surgiram no século I depois de Cristo, contemporaneamente às clepsidras, ou relógios de água. Chegou-se a desenvolver ampulhetas bastante grandes, com ciclos de até uma hora. Nos séculos XVI e XVII eram comuns ampulhetas de 15 e 3Ø minutos nas igrejas, usadas para marcar o tempo dos sermões. Para melhor medir o tempo foram feitas ampulhetas com as metades superiores e inferior cilíndricas, de tal modo que a altura da areia fosse proporcional ao tempo passado, e portanto pudesse haver marcas de minuto em minuto, até um máximo de vinte minutos. Muitas ampulhetas tinham um eixo no centro, para que ao fim de um ciclo, se pudesse rodá-las e começar de novo.

O Programa

Linhas 118Ø-123Ø – Essa parte do programa POKEeia uma subrotina em linguagem de máquina, para simular o som da areia caindo.

Linhas 25Ø-27Ø – "Perguntam" quanto tempo o programa deve cronometrar e o local de origem da areia. Perceba que para cada tipo de areia há um valor de **D** e **X**, que serão usados na rotina de som, para produzir sons diferentes para cada tipo de areia. Sinta-se à vontade para mudar os locais de origem das areias, mudando também, conforme necessário, as linhas 28Ø-31Ø.

Linhas 33Ø-98Ø – Formam o loop FOR-NEXT que determina o tempo a ser cronometrado.

Linhas 39Ø-47Ø – Controem a ampulheta em si.

Linhas 49Ø-62Ø – Enchem a metade superior com areia de cor aleatória.

Linhas 63Ø-74Ø – Formam a rotina que tira um grão de areia da metade de cima e coloca na metade de baixo. Como? Simplesmente pintando o ponto da parte de cima de preto e o ponto da parte de baixo da cor anterior. As posições dos pontos são lidas através das linhas 67Ø e 71Ø.

Linhas 75Ø-77Ø – Constituem um **loop** para passar o tempo, de tal forma que cada grão caia em intervalos de 1 segundo. Para um ajuste fino do tempo, talvez seja preciso mudar o valor final deste **loop** (um valor maior tornará a execução mais lenta, enquanto um valor menor a tornará mais rápida). O valor 22Ø funciona bem no nosso computador.

Linhas 78Ø-88Ø – Dão a ilusão da areia caindo, PLOTando e apagando pontos consecutivos suficientemente rápido para que eles pareçam estar se movendo. Use esta técnica em seus programas. A linha 87Ø chama a subrotina em linguagem de máquina para fazer o som. Para cada tipo de areia há um som específico. Se você quiser, remova o REM dessa linha para que ela fique:

87Ø POKE 768,F + NS: POKE 769,S: CALL 77Ø:NS = NS + 1

e então para cada grão que cair, haverá um novo som.

Linhas 89Ø-94Ø – Toca o sininho tantas vezes quanto os minutos que já passaram.

Linhas 1ØØØ-116Ø – Formam a areia, isto é, as coordenadas dos pontos que devem ser apagados e pintados pela rotina das linhas 63Ø-74Ø.

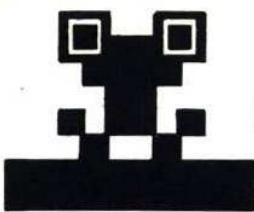
TWO-LINERS

Ainda estamos esperando suas colaborações para apreciação de nossa redação e posterior publicação.

Portanto, mãos à obra e enviem seus programas para: **MICROHOBBY**
Seção Por Dentro do Apple
Caixa Postal 60081

```
Ø REM   *** AMPULHETA ***

Ø1983 POR DENTRO DO APPLE

100 TEXT : HOME : CLEAR
240 GOSUB 1180
250 INPUT "QUANTOS MINUTOS DEVO
     CRONOMETRAR ? ";MI
260 HOME : VTAB 7
270 PRINT "QUE TIPO DE AREIA DEV
     O USAR ?": PRINT : PRINT "DE
     SERTO DO (M)OHAVE": PRINT "D
     ESERTO DE (A)TACAMA": PRINT
     "DESERTO DE (G)OBI": PRINT "
     DESERTO DO (S)AARA": PRINT :
      PRINT "SUA ESCOLHA ? ";: GET
     A$: PRINT A$
280 IF A$ = "M" THEN D = 25:X =
     10
290 IF A$ = "A" THEN D = 50:X =
     10
300 IF A$ = "G" THEN D = 50:X =
     5
310 IF A$ = "S" THEN D = 100:X =
     5
320 HOME
330 FOR TP = 1 TO MI
340 GR
350 R = 37
360 Z =  INT (14 *  RND (1)) + 1
370 COLOR= 15
390 REM CONSTRUIR AMPULHETA
410 HLIN 13,27 AT 0: HLIN 13,27 AT
     39
420 VLIN 1,7 AT 13: VLIN 32,39 AT
     13
430 VLIN 1,7 AT 27: VLIN 32,39 AT
     27
440 S = 14:J = 21
450 FOR A = 1 TO 12 STEP 2: PLOT
     S,A + 7: PLOT S,A + 8: PLOT
     J,A + 19: PLOT J,A + 20:S =
     S + 1:J = J + 1: NEXT A
460 S = 14:J = 21
470 FOR A = 12 TO 1 STEP  - 2: PLOT
     S,A + 18: PLOT S,A + 19: PLOT
     J,A + 6: PLOT J,A + 7:S = S +
     1:J = J + 1: NEXT A
480 REM
490 REM ENCHER A METADE DE CIMA
500 REM
510 COLOR= Z
520 A = 17:C = 19:B = 21
530 FOR K = 1 TO 2
540 FOR J = C TO B
550 PLOT J,A
560 NEXT J
570 A = A - 1
580 NEXT K
590 C = C - 1:B = B + 1
600 IF A < 6 THEN 620
610 GOTO 530
620 HLIN 14,26 AT 38
630 REM
640 REM COMECA A MARCAR O TEMPO
650 REM
660 COLOR= 0
670 READ X,Y
680 IF X = 0 THEN 900
690 PLOT X,Y
700 COLOR= Z
710 READ X,Y
720 PLOT X,Y
730 IF X = 20 THEN R = R - 1
740 REM
750 REM AJUSTA O RELOGIO
760 REM
770 FOR K = 1 TO 220: NEXT K
780 REM
790 REM CAIR DA AREIA
800 REM
810 FOR F = 19 TO R
820 COLOR= Z
830 PLOT 20,F
840 COLOR= 0
850 PLOT 20,F - 1
860 NEXT F
870 POKE 768,D + NS: POKE 769,X:
      CALL 770: REM OPCAO...NS=NS
     +1
880 GOTO 660
890 REM
900 REM TEMPO QUE JA PASSOU
910 REM
920 FOR S = 1 TO TP
930 PRINT  CHR$ (7)
940 NEXT S
950 IF S = 2 THEN  HOME : HTAB 1
     8: PRINT "1 MINUTO": GOTO 97
     0
960 HOME : HTAB 17: PRINT TP" MI
     NUTOS"
970 RESTORE :NS = 0
980 NEXT TP
990 END
1000 REM
1010 REM DADOS DA AREIA
1020 REM
1030 DATA 20,6,20,37,19,6,19,37
     ,21,6,21,37,20,7,20,36,19,7,
     19,36,21,7,21,36
1040 DATA 20,8,20,35,18,6,18,37,
     22,6,22,37,20,9,20,34,19,8,1
     9,35,21,8,21,35,21,9,22,36
1050 DATA 22,7,22,35,22,8,23,37,
     23,6,21,34,21,10,20,33,20,10
     ,19,34,19,9,18,36,22,9,18,35
1060 DATA 20,11,17,37,18,7,17,3
     6,17,6,16,37,23,7,20,32,21,1
     1,21,33,22,10,23,36,16,6,24,
     37
1070 DATA 16,6,18,34,20,12,17,35
     ,18,8,17,34,17,7,19,33,17,8,
     18,33,24,6,15,37,19,10,16,36
1080 DATA 16,7,23,35,18,9,19,32,
     23,8,16,35,24,7,24,36,25,6,2
     2,34,21,12,25,37,23,9,22,33
1090 DATA 19,11,20,31,23,10,23,3
     4,24,8,26,37,22,11,26,36,18,
     10,21,32,22,12,25,36
1100 DATA 20,13,24,35,21,13,16,3
     4,15,7,15,36,15,6,18,32,15,8
     ,15,35,14,6,14,37,24,10,17,3
     3
1110 DATA 14,7,25,35,16,8,24,34
     ,17,9,18,31,24,9,19,31,19,12
     ,20,30,20,14,26,35,25,7,22,3
     2
1120 DATA 25,8,14,36,26,6,21,31,
     26,7,25,34,19,13,19,30,20,15
     ,17,32,21,14,21,30,18,11,23,
     33
1130 DATA 17,10,16,33,16,9,15,34
     ,25,9,14,35,15,9,24,33,22,13
     ,22,31,16,10,20,29,17,11,23,
     32
1140 DATA 18,12,26,34,23,11,17,3
     1,24,11,16,32,21,15,21,29,19
     ,14,14,34,23,12,20,28
1150 DATA 23,13,15,33,16,11,23,
     31,22,14,18,30,18,13,25,33,1
     7,13,22,30,22,15,24,32,17,12
     ,19,29
1160 DATA 18,14,18,29,20,16,21,2
     8,19,15,19,28,21,16,22,29,18
     ,15,17,30,20,17,24,31,19,16,
     16,31,21,17,23,30,19,17,20,2
     7,0,0
1180 REM SOM DA AREIA
1200 POKE 770,173: POKE 771,48: POKE
     772,192: POKE 773,136: POKE
     774,208: POKE 775,5: POKE 77
     6,206: POKE 777,1
1210 POKE 778,3: POKE 779,240: POKE
     780,9: POKE 781,202: POKE 78
     2,208: POKE 783,245: POKE 78
     4,174: POKE 785,0: POKE 786,
     3
1220 POKE 787,76: POKE 788,2: POKE
     789,3: POKE 790,96
1230 RETURN
```

Daniel R. Falconer e José Eduardo Moreira
Professores assistentes do
Departamento de Computação
da Escola Pueri Domus.

A FITA DO MÊS:

O PULO DO SAPO

Nos números anteriores já analisamos um jogo inteligente (TK Xadrez de MICROSOFT); um jogo sério (Simulador de Vôo — fita brinde de MICROHOBBY); um programa-ferramenta (funções I da MICRON) e um aplicativo administrativo (SICOM da MICROSOFT). Este mês vamos analisar um jogo de ação: O PULO DO SAPO da MULTISOFT.

A fita nos foi enviada numa embalagem bem cuidada, contendo instruções de carga e um certificado de garantia. Já aqui, nos defrontamos com a primeira novidade: a fita está garantida por 60 dias "contra defeitos de fabricação" e por UM ANO, contra "uso indevido ou mau funcionamento do equipamento reprodutor mediante pagamento de uma taxa de substituição".

Se realmente isto funcionar, representa uma louvável iniciativa que esperamos seja logo imitada por todos os fabricantes brasileiros de SOFTWARE.

Afinal das contas o tão massacrado consumidor precisa começar a levar um pouco de vantagem.

A qualidade de gravação foi testada aqui, na redação, em gravadores de várias marcas e modelos e justificou a ousadia de garantia. A gravação (bem uniforme e em bom nível) só dá problemas em gravadores com o azimuth da cabeça desregulado. Este é um problema que ainda esta à espera de uma solução.

O PULO DO SAPO é um jogo tipo "vídeo-game" que pode ser jogado com Joystick. Obviamente não resiste a uma comparação direta com cartuchos de vídeo-game por três motivos básicos:

1) No cartucho, o tempo de carga é irrelevante (o cartucho é uma ROM.). Na fita em questão, o tempo de carga é de 6 minutos.
2) A tela do TK é de baixa resolução (pixel muito grande) para poder permitir figuras "realistas".
3) O TK só produz imagens preto e branco.

Mas, levando-se em conta que o TK é um computador (de verdade!) e que esses jogos representam nele, uma utilidade marginal, talvez as objeções anteriores possam ser relevadas.

Nossa opinião começa a se tornar um pouco mais favorável quando começamos a jogar: a apresentação está em nível profissional e as instruções são bem nítidas (em que pese a ausência de acentos!).

O conteúdo do jogo é um pouco pobre: um sapo que tenta chegar em casa atravessando uma estrada movimentada e pulando por cima de troncos, tartarugas e jacarés para atravessar um rio.

Um toque patético é dado pelo esporádico aparecimento de um filhote (ou será um girino?) que o sapo precisa salvar. Quando o sapo pega o filhote no colo, o símbolo (não dá para se falar em figuras) pisca com um rítmo tão bem acertado que chega a evocar algo esperneando! O efeito é gozadíssimo, mostrando que a criatividade e a imaginação podem muito bem suprir a falta de recursos.

A contagem de pontos e de tempo é impecável e o sapo dispõe, inclusive, de uma galeria de recordes, bem estimulante para o espírito competitivo.

Quando um jogador escreve seu nome, se o joystick estiver conectado, algumas letras não são aceitas. Este problema ocorre com alguns modelos antigos do TK 82-C e é de máquina,mas não de fita (aliás já comentamos, na revista, como o usuário deve proceder para eliminar este inconveniente).

Em resumo, trata-se de uma fita produzida e fabricada em nível profissional, contendo um jogo razoavelmente interessante, com pouco conteúdo instrutivo mas suficientemente divertido para um dia de chuva ou uma reunião de amigos.

(N.R.: **Todos** os fabricantes de SOFTWARE podem nos enviar um exemplar de suas fitas, pois teremos o máximo prazer em analisá-las e comentá-las para prestar mais um serviço a nossos leitores).

NOVIDADES

O TK 2000

As principais características deste lançamento da Microdigital são:

- **64K de memória RAM**
- **16K de memória ROM com linguagem BASIC (compatível com Apple Soft), Mini-Assembler e Monitor/Disassembler.**
- **Operação em modo texto ou gráfico de baixa e alta resolução (280x192), em cores, sistema PAL-M.**
- **Canal de som pela TV.**
- **Teclado profissional tipo máquina de escrever.**

Podem ser conectados ao TK2000 até dois gravadores cassete com controle por software, em formato Apple-Soft ou próprio, e também pode ser conectado interface para disk-drive de 5 1/4 polegadas.

Para display de vídeo pode ser utilizado numa TV comum, em cores ou preto e branco, ou monitor de vídeo.

A ligação de impressora, tipo Centronics, pode ser feita diretamente, sem a necessidade de dispositivos de interface adicionais.

Está previsto o lançamento de outros acessórios tais como joystick, interface RS232, etc.

Uma grande quantidade de programas aplicativos comerciais e profissionais, bem como jogos animados em cores e alta resolução será lançada.

Na área de aplicativos comerciais destacam-se:

1) programa de cálculo de planilhas (tipo Visicalc),
2) sistema de controle de estoques,
3) editor de textos para a lingua portuguesa.

NOVO LIVRO PARA INICIAÇÃO AO MICROCOMPUTADOR

Está sendo lançado no Rio de Janeiro, um novo livro que visa difundir a linguagem básica para microcomputadores, em todo território nacional e nos países de língua portuguesa.

O livro, que serve para adultos e crianças é fruto de um método e a experiência de um curso para iniciantes, desenvolvido pela MICRO-KIT, tem como autores Carlos Alberto Abreu, que possui mestrado em Engenharia de Sistemas pela COPPE e de Denise Reis, que também possui mestrado em Educação do IESAE, da Fundação Getúlio Vargas.

Tendo como base a programação voltada para os microcomputadores da linha Sinclair: TK82, 83, 85, CP200, TIMEX 1000, ZX81, entre outros; esse livro pode também ser utilizado pelos operadores da linha TRS ou APPLE, com ligeiras modificações.

Sendo editado no Rio de Janeiro, o livro será distribuído diretamente pela Micro-Kit em sua primeira edição.

PROGRAMAS PARA COMPATÍVEIS COM TRS-80

Foi lançado livro de programas para CP300, CP500, D8000, D8001, D8002, DGT100, NAJA, JR.SYSDATA de autoria de Jeff Andrew editado pela EB Informática (CP. 30234, CEP 1051, SP).

Os programas varrem toda gama de interesse, indo desde aplicativos sérios (conta bancária, demonstrativos gráficos, etc.) até jogos divertidos.

A INFORMÁTICA CHEGA ÀS ESCOLAS DE 1º E 2º GRAUS

Crianças e adolescentes de várias escolas particulares do 1º e 2º graus, já estão aprendendo a operar em microcomputadores pessoais, sendo esse o primeiro passo para que, no futuro, eles possam desenvolver-se no promissor campo da informática.

O curso, que está sendo ministrado pela CENADIN — Centro Nacional do Desenvolvimento da Informática, sob a direção do engenheiro Ramiro Yabumoto e Alexandre L. Werfel, seguem as seguintes fases: Iniciação ao Microcomputador — Programação Basic e Introdução ao Processamento de Dados.

Com um corpo de vários professores especializados, esta empresa ministra esse curso no próprio colégio, onde, em razão do próprio interesse dos pais dos alunos, lhes foi cedido um espaço para essa especialização futura.

Colégios de boa categoria, como o Rio Branco, Renascença, Benjamin Constant, Maria Imaculada entre outros, já cederam espaços para que a Cenadin possa iniciar seus alunos no campo da informática.

A iniciativa pioneira desta empresa, acha-se em franca expansão e apesar da resistência de algumas escolas, o engenheiro Alexandre acredita que outros colégios de São Paulo irão aderir, uma vez que, o próprio avanço da informática irá determinar esse procedimento.

CURSO DE ASSEMBLY
aula 4

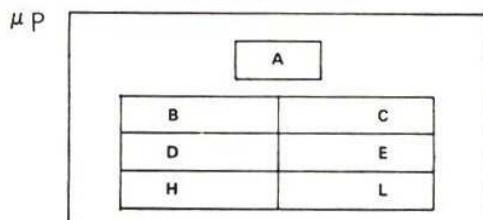
Flavio Rossini

Observação Inicial: Para esclarecimento dos leitores, salientamos que este curso é um resumo do livro *"Introdução à Linguagem de Máquinas" para o TK" — ASSEMBLY Z80",* Volume I de Flávio Rossini (Editora Moderna e Micromega). Para os que desejarem se aprofundar no ASSEMBLY Z80, sugerimos a leitura do livro citado e sua continuação (Volume II).

PRINCIPAIS REGISTROS INTERNOS DE 8 BITS

Iniciaremos esta aula explorando o conceito de REGISTRO; esta palavra já foi utilizada quando falamos sobre a **memória.** Um **registro** é um circuito eletrônico capaz de memorizar **bits** (**oito** no nosso caso), podendo representar e armazenar números de Ø a 255 ('ØØ' a 'FF' em hexadecimal).

O microprocessador do TK possui vários registros **internos** que são muito utilizados nos programas em linguagem de máquina, dos quais sete serão estudados nessa aula; estes registros "não" fazem parte da memória e são chamados respectivamente de A, B, C, D, E, H e L. Alguns deles têm a propriedade de "juntarem-se" quando necessário, formando um único registro de 16 **bits;** assim podemos formar os pares de registros BC, DE e HL (só estes pares são possíveis).

Por exemplo, se o registro B contém o número 'AØ' (16Ø em decimal) e o registro C contém 'B2' (178 em decimal) o par BC contém o número 'AØB2' (256 x x 16Ø + 178 = 41138 em decimal).

(Como podem ser imaginados os principais registros internos do microprocessador)

Como vimos anteriormente, as instruções em linguagem de máquina são dadas por um ou mais grupos de códigos em hexadecimal de dois dígitos cada. Isto é feito para evitar a confusão que poderia ser criada usando apenas os números Ø e 1. Mesmo assim, temos agora apenas 16 símbolos disponíveis (Ø, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F) e para programas razoavelmente grandes mesmo isto pode trazer confusão! Dessa forma, para facilitar a compreensão do ponto de vista humano, costuma-se associar a cada instrução, grupos de letras em forma MNEMÔNICA para auxiliar a programação, que são chamados "códigos de operação", pois terão como objeto associar uma **ORDEM** a cada grupo de letras! Por exemplo, se o código **1Ø1ØØ111** significar "some", seu correspondente em hexadecimal seria **'A7'** e seu **mnemônico** poderia ser **ADD.** Assim, dependendo da instrução em linguagem de máquina, cada mnemônico poderá significar 1 ou mais grupos de 2 dígitos hexadecimais. Dessa maneira, para programar em linguagem de máquina, deveremos aprender os mnemônicos correspondentes a cada instrução do microprocessador e fazer o programa utilizando os mesmos; no final com o auxílio de uma tabela, substituiremos os mnemônicos por seus códigos hexadecimais e utilizaremos o HEXAMEM para colocar o programa em **bits** no computador.

No TK os programas em linguagem de máquina são tratados como SUBROTINAS de um programa em BASIC, as quais são chamadas pela função **USR** (em BASIC, abreviação de USER) cujo argumento é o **endereço inicial da subrotina** (por exemplo **3ØØØØ** se o programa de linguagem de máquina estiver a partir do endereço que colocamos em **RAMTOP** em aulas anteriores); portanto, estas subrotinas devem ter como instrução final um comando para voltar ao BASIC, similar ao **RETURN,** cujo código hexadecimal e mnemônico são respectivamente:

'C9' (código) **RET** (mnemônico abreviação de RETURN)

Vamos entender como funciona a instrução USR; suponhamos que tivéssemos o seguinte "programa" em BASIC:

10 PRINT USR 30000 (USR está na tecla L do TK)

Ao encontrar **USR 30000,** o computador "carrega" o par de registros BC (mencionados anteriormente) com este número, ou seja, como **30000** = **'7530',** teremos B = '75' e C = '30'. A seguir o microprocessador executa o programa em linguagem de máquina que começa na memória **30000** (= **'7530'**); portanto é conveniente que **tenha sido previamente colocado um programa** a partir da memória **30000.** O programa é então executado até encontrar a instrução **RET** (código **'C9'**) quando o controle volta ao programa BASIC, fornecendo como saída o conteúdo do par BC, que, se não for alterado durante o programa em linguagem de máquina, continuará sendo **'7530' (30000).** Portanto, para obter a saída da subrotina (ou seja, os resultados que devem estar no par BC) basta ordenar um **PRINT** da função USR!

Vamos então colocar o seguinte programa em linguagem de máquina na memória **30000: (que já deve estar reservada!)**

RET (apenas isto!)

Como se trata apenas de uma instrução, não precisamos utilizar o programa HEXAMEM; iremos fazê-lo diretamente com POKE. O código hexadecimal do RET é 'C9' que equivale a 201 (em decimal), portanto:

POKE 30000,201 **(NEW LINE)**

A seguir digite:

PRINT USR 30000 **(NEW LINE)**

O que deverá aparecer na tela? Ora, a função USR carregará o par BC com 30000 ('7530'), ordenando a seguir, que o microprocessador execute o programa em linguagem de máquina que está a partir da memória 30000, encontrando "de cara" o RET. Dessa forma, o controle voltará ao programa em BASIC, sem ter alterado os registros BC e fornecerá como saída, o próprio **30000;** portanto, na tela de TV deverá aparecer o número **30000.**

Poderia surgir agora a seguinte pergunta: mas não existe uma instrução equivalente ao STOP? De fato, há em linguagem de máquina, uma instrução similar ao STOP em BASIC cujo código de operação é **HALT** (código hexadecimal **'76'**). **NUNCA** use esta instrução pois, **neste computador,** ela paralisa o sistema e este não aceita mais nenhuma tecla nem mesmo o **BREAK!** Portanto, use sempre RET para terminar seus programas em linguagem de máquina.

A INSTRUÇÃO LD

A instrução **LD** permite colocar números dentro dos registros internos do microprocessador (obviamente estes números poderão ser de **'00'** a **'FF'** no máximo para cada registro), copiar os dados de um registro em outro

ou para algum endereço da memória e vice-versa; assim, existirão várias instruções LD, cada uma com seu próprio código hexadecimal. (**LD** é abreviatura de LOAD = = carregue).

Vamos analisar inicialmente, como colocar números diretamente nos vários registros; a instrução, de modo genérico, corresponde a:

LD registro,dado

Por exemplo:

LD E, '2A' (que equivale a dizer LD E,42)

Colocará o número '2A' (42) no registro E. Esta instrução corresponde a 2 (dois) **bytes** em código hexadecimal, um para o código de operação (no caso "carregue" o registro E: LD E,) e o outro para o dado número que será colocado. Assim, para os vários registros, teremos os seguintes códigos de operação e códigos hexadecimais:

INSTRUÇÃO	CÓDIGO		
LD A, dado	'3E'	+	1 byte para o dado
LD B, dado	'06'	+	1 byte para o dado
LD C, dado	'0E'	+	1 byte para o dado
LD D, dado	'16'	+	1 byte para o dado
LD E, dado	'1E'	+	1 byte para o dado
LD H, dado	'26'	+	1 byte para o dado
LD L, dado	'2E'	+	1 byte para o dado

(a instrução LD registro,dado (8 bits).)

continua

Vamos então fazer um programa, que "carrega" o registro B com 'ØØ' e o registro C com '2A':

```
MEM.30000   LD B,'00'    '0600'   ;'00' =  0
MEM.30002   LD C,'2A'    '0E2A'   ;'2A' = 42
MEM.30004   RET          'C9'
```

Devemos então colocá-lo, por exemplo, a partir da memória 3ØØØØ (lembre-se de reservá-la), usando 5 vezes POKE como fizemos anteriormente para a instrução RET (transformando adequadamente os códigos hexadecimais para decimais) ou então usando o programa HEXAMEM para introduzir as **3 instruções.** Se você utilizar o HEXAMEM, lembre-se de RESERVAR o fim da memória **antes** de colocá-lo no computador; a seguir, execute-o colocando 3ØØØØ para o endereço inicial e faça: (N.L. = NEW LINE)

```
0600     (N.L.)
0E2A     (N.L.)
C9       (N.L.)
P        (N.L.)
```

Note que o programa ocupa 5 posições de memória, respectivamente de 3ØØØØ a 3ØØØ4.

```
30000 | 00000110 | '06'
                         LD B, '00'
30001 | 00000000 | '00'
30002 | 00001110 | '0E'
                         LD  C,  '2A'
30003 | 00101010 | '2A'
30004 | 11001001 | 'C9'  RET
30005 | ........ |
```

(visualização do programa na memória do computador)

A seguir ditite:

PRINT USR 3ØØØØ (N.L.)

O que deverá aparecer na tela? O conteúdo do par de registros BC, ou seja, 256 x Ø + 42 = 42. Isto porque o par de registros foi alterado antes da instrução RET; assim seu conteúdo inicial que era BC = 3ØØØØ ('753Ø) foi modificado pelo programa para BC = 42 ('ØØ2A').

Há também a possibilidade de carregar diretamente **pares de registros;** neste caso porém, cada instrução corresponderá a 3 **bytes,** 1 para o código de operação e 2 para o número que deverá ter 16 **bits.** Note que, como já foi comentado anteriormente, você deverá colocar **antes SEMPRE O BYTE MENOS SIGNIFICATIVO;** assim temos as seguintes instruções:

INSTRUÇÃO	CÓDIGO
LD BC, dado	'Ø1' + 2 bytes para o dado
LD DE, dado	'11' + 2 bytes para o dado
LD HL, dado	'21' + 2 bytes para o dado

(a instrução LD par de registros, dado (16 bits).)

E façamos o programa:

```
MEM. 30000  LD BC, '00A0'  '01A000'  ; inversão de
                                       'A0' com '00'
MEM. 30002  RET            'C9'
```

(veja nº 4 ou nº extra-promocional)

Usando HEXAMEM, coloque o programa a partir da memória 3ØØØØ. Verifique que fazendo PRINT USR 3ØØØØ você obterá 16Ø (Por quê?). Você não precisa preocupar-se em "apagar" o programa anterior, pois estará escrevendo "por cima" nas mesmas memórias.

Experimente "esquecer" de inventar os **bytes** conforme indicado e introduza este programa:

```
'0100A0'
'C9'
```

O que você irá obter? Tente explicar o resultado.

Vamos ver agora, como passar números de um registro para o outro; teremos todas as combinações possíveis da seguinte instrução genérica:

LD registro, registro,

a qual copia o conteúdo do registro da DIREITA no registro da ESQUERDA, **não alterando** o conteúdo do registro da direita; esta instrução equivale a apenas **1 byte** em código hexadecimal pois não exige que nenhum dado seja explicitado. Para facilitar, iremos construir uma tabela que possui todas as combinações possíveis desta instrução:

LD	A	B	C	D	E	H	L
A	'7F'	'78'	'79'	'7A'	'7B'	'7C'	'7D'
B	'47'	'4Ø'	'41'	'42'	'43'	'44'	'45'
C	'4F'	'48'	'49'	'4A'	'4B'	'4C'	'4D'
D	'57'	'5Ø'	'51'	'52'	'53'	'54'	'55'
E	'5F'	'58'	'59'	'5A'	'5B'	'5C'	'5D'
H	'67'	'6Ø'	'61'	'62'	'63'	'64'	'65'
L	'6F'	'68'	'69'	'6A'	'6B'	'6C'	'6D'

Para utilizá-la, use sempre primeiro a coluna vertical

à esquerda e, a seguir, a linha horizontal superior; por exemplo:

LD C,D equivale a **'4A'** (copie o conteúdo do registro D no registro C, sem alterar o registro D).

LD E,E equivale a **'5B'** (copie o conteúdo do registro E no registro E. Como você pode notar esta instrução não faz absolutamente **nada**!).

Experimente agora o seguinte programa, que copia o conteúdo dos registros **H** e **E,** previamente carregados, nos registros B e C respectivamente: (lembre-se que, por enquanto, iremos fazer sempre **MEMÓRIA INICIAL** = = **3ØØØØ** e que o final de memória **deve** estar reservada antes de colocar HEXAMEM no computador).

```
MEM. 3ØØØØ  LD H,'Ø1'   '26Ø1'  ; carrega H com 'Ø1'
MEM. 3ØØØ2  LD E,'5Ø'   '1E5Ø'  ; carrega E com '5Ø'
MEM. 3ØØØ4  LD B,H      '44'    ; copia H em B
MEM. 3ØØØ5  LD C,E      '4B'    ; copia E em C
MEM. 3ØØØ6  RET         'C9'    ; volta ao BASIC
```

Utilize então **HEXAMEM** para colocar os códigos do programa na memória e, no final, ao invés de teclar **P** (pare), tecle **XS** (e**X**ecute em **S**low) ou **XF** (e**X**ecute em **F**ast) para que o programa seja executado. Naturalmente se você prefere continuar fazendo PRINT USR 3ØØØØ não há problemas.

Perceba que este programa "carrega" o registro **H** com 'Ø1' e o registro E com '5Ø' (8Ø em decimal) e, a seguir, copia o registro H em B e o registro E em C, fazendo BC = 'Ø15Ø'. Desse modo, você poderá obter 1 * 256 + + 8Ø = 336.

(NOTA: NÃO existe instrução para copiar de uma só vez números de 1 par de registros para outro; assim, por exemplo, a instrução LD BC, HL **não** é válida e deve ser substituída por:

LD B,H e **LD C,L**

Além disso, NÃO é possível tentar copiar o conteúdo de um registro para um par ou vice-versa; portanto, instruções do tipo:

LD BC,A
LD D,HL não existem!

A título de esclarecimento, podemos agora fazer uma **analogia** com o BASIC; a instrução LD registro, dado de 1 byte ou LD par de registros, dado de 2 bytes pode ser associada à instrução LET variável = número, por exemplo, **LD B, '1Ø'** com **LET X = 16.** A instrução **LD** registro, registro pode ser associada a **LET variável** = **variável,** por exemplo, **LD H,L** com **LET X = Y.**

É conveniente lembrar que ao chamar uma subrotina em linguagem de máquina, **apenas o conteúdo dos registros B e C é conhecido;** todos os demais têm um conteúdo desconhecido. O mesmo é válido para a memória que fica **após o programa.** Vamos exemplificar: ao fazer **RAMTOP** = **3ØØØØ,** reservamos a memória de **3ØØØØ** a **32767.** Se tivermos um programa que ocupa **1ØØ** bytes, ou seja, da memória **3ØØØØ** à memória **3ØØ99,** as posições de **3Ø1ØØ** a **32767** terão valores "quaisquer". É nesta região de memória **APÓS** o programa em linguagem de máquina que colocaremos as **variáveis** do nosso programa que não "couberem" nos registros. ○

OS OITENTA

UM EDITOR DE CARTAS

Igor Sartori

Um dos recursos mais interessantes de um computador é sua capacidade de manipulação de texto. Ou seja, juntar, separar, quebrar e classificar ***strings*** (cadeias). Isso permite usar o computador e sua impressora como uma máquina de escrever "inteligente".

Programas com esta finalidade, são chamados de "editores" ou "processadores de texto" e atualmente, são bastante usados em jornais para facilitar os trabalhos de diagramação, revisão e redação de artigos.

Em algumas revistas especializadas têm aparecido muitos programas que realizam uma ou outra função de manipulação de ***strings,*** principalmente para computadores compatíveis com o TRS-80 e o Apple II.

Se você tem um TK-82 e viu algum destes programas, deve ter pensado: "Interessante . . . mas como eu poderei fazer o mesmo no meu computador?"

Um editor de cartas

Mudando um pouco o enfoque dado em artigos anteriores, mostraremos um programa destinado a outros computadores e depois de explicarmos detalhadamente seu funcionamento, descreveremos como você poderá implementar um programa semelhante em seu TK.

Este programa (veja listagem) tem a finalidade de colocar um texto de uma correspondência num formato de carta padrão. O texto deverá entrar linha por linha, poderá ser corrigido em qualquer momento de sua digitação — sendo impresso após a digitação da última linha, limpo e sem erros, já que eles foram eliminados no processo de escrita. Foi desenvolvido em BASIC padrão, podendo ser rodado na maioria dos computadores, mediante adaptações à sua respectiva versão do BASIC (por exemplo, o Apple II e o TRS-80 e seus similares).

```
10 '    EDITOR DE CARTAS
20 CLEAR 2000
30 DIM A$(100)
40 INPUT "QUAL O NUMERO MAXIMO DE CARACTERES";N
50 INPUT "NOME DO AUTOR DA CARTA";N$
60 INPUT "DATA (POR EXTENSO) ";D$
70 INPUT "CIDADE DO AUTOR DA CARTA ";L$
80 INPUT "NOME DO DESTINATARIO ";D1$
90 INPUT " ENDERECO (Rua e numero) DO DESTINATARIO ";E$
100 INPUT "CEP, O NOME DA CIDADE, O ESTADO DO DESTINATARIO (EM UMA UNI
CA LINHA)";T1$
110 INPUT "DIGITE 'Prezado(a) Sr(a.) PRIMERO NOME DO DESTINATARIO'";A1
$
120 PRINT CHR$(31)
130 PRINT"INSTRUCOES"
140 PRINT "INTRODUZA AS LINHAS UMA A UMA."
150 PRINT "SE OCORRER UM ERRO EM UMA DELAS,ELA PODERA SER"
160 PRINT "SER CORRIGIDA DIGITANDO-SE  '#' E SEU RESPECTIVO NUMERO"
170 PRINT"DIGITE 'FIM' PARA PARAR"
180 FOR X=1 TO 100
190 Z=X
200 PRINT "#";X
210 INPUT A$(X)
220 IF LEFT$(A$(X),1)="#" THEN 450
230 IF A$(X)="FIM" THEN PRINT CHR$(31):GOTO 260
240 IF LEN(A$(Z)) > N THEN PRINT "LINHA MUITO GRANDE":Y=Z:X=X+1:C=1:M=
0:GOTO 470
250 NEXT X:X=X-1
260 PRINT TAB(5);L$;",";D$
270 PRINT:PRINT:PRINT:PRINT
280 PRINT TAB(5);D1$
290 PRINT TAB(5);E$
300 PRINT TAB(5);T1$
310 PRINT
320 PRINT TAB(5);A1$;":"
330 PRINT
340 FOR B=1 TO X-1
350 PRINT TAB(5);A$(B)
360 NEXT B
370 C=23+B
380 PRINT
390 PRINT
400 PRINT TAB(55);"Atenciosamente"
410 PRINT
420 PRINT
430 PRINT TAB(55);N$
440 END
450 '        ROTINA DE CORRECAO
460 Y=VAL(RIGHT$(A$(X),LEN(A$(X))-1))
470 INPUT "SELECAO; 1)INTRODUZIR NOVAMENTE A LINHA 2)CORRECAO";A
480 IF A<1 OR A>2 THEN 470
490 ON A GOTO 500,520
500 A$(Y)="":INPUT A$(Y)
510 X=X-1:IF LEN(A$(Y))<=N OR A$(Y)="" THEN GOTO 250 ELSE LET Z=Y:GOTO
 240
520 T=LEN(A$(Y))
530 C=1
540 LET M=0
550  Q$=INKEY$
560 IF Q$=" " THEN IF C>T THEN  GOTO 550 ELSE LET M=M+1:C=C+1:I$=MID$(
A$(Y),M,1):PRINT I$;
570 IF Q$="D" AND C<=T THEN A$(Y)= LEFT$(A$(Y),C-1)+ RIGHT$(A$(Y),LEN
(A$(Y))-C)
580 IF Q$="I"  THEN  INPUT J$:A$(Y)=LEFT$(A$(Y),C-1)+J$+RIGHT$(A$(Y),L
EN(A$(Y))-C+1)
590 IF Q$="E"  THEN IF LEN (A$(Y))<=N  THEN PRINT :PRINT A$(Y):A$(X)="
":X=X-1:GOTO 250:ELSE LET X=X-1:C=1:M=0:Z=Y:PRINT:PRINT A$(Y):GOTO 240
600 IF Q$="L"  THEN PRINT RIGHT$(A$ (Y),(LEN(A$(Y))-C+1)):C=1:M=0
610 IF Q$="B" THEN PRINT CHR$(18);:LET M=M-1:C=C-1
620  GOTO 550
```

FUNCIONAMENTO

O programa **editor de cartas** está dividido em quatro partes: introdução dos dados do autor da carta e do destinatário, introdução do texto da carta, correção dos possíveis erros e impressão da carta.

Na primeira parte, que compreende as linhas de 10 a 120, reserva-se inicialmente um espaço na memória para o texto da carta, por meio das instruções CLEAR 2000 e DIM A$(100). A instrução **CLEAR 2000** reserva 2000 bytes da memória para **strings** e a instrução **DIM A$(100)** dimensiona um vetor de 100 posições de memória, que conterá o texto da carta. Desta forma, poderemos escrever uma carta com no máximo 100 linhas ou 2000 letras (cerca de 200 a 300 palavras), o que é suficiente para um bom "bate-papo".

A seguir, dimensiona-se o número máximo de caracteres por linha. Isso será função do tamanho do papel disponível e do tamanho da margem direita e esquerda. No programa, a margem esquerda foi fixada em 5 espaços, ficando a margem direita a critério do usuário.

A seguir são pedidos os dados do autor da carta e do destinatário. Na linha 100 pede-se o **CEP,** a cidade e o Estado do Destinatário. Estes dados deverão ser digitados em uma única linha, separados por um espaço.

Uma vez digitados todos os dados, a tela é limpa, por meio da instrução **PRINT CHR$(31).** Isso prepara a tela do computador para a introdução do texto da carta.

A INTRODUÇÃO DO TEXTO

No momento da introdução do texto, o programa mostra na tela uma série de instruções destinadas a explicar como proceder.

A seguir, o computador mostra o símbolo "#" seguido pelo número 1. Isso indica que você deve digitar a primeira linha de seu texto. Quando terminar de digitá-la, pressione a tecla **ENTER, NEW LINE** ou equivalente. Se a linha for muito grande, o programa entra automaticamente no modo de correção, exibindo a mensagem:

LINHA MUITO GRANDE

Se, por outro lado, o número de caracteres digitados for menor que aquele estabelecido no início do programa, o computador exibirá novamente o símbolo "#", seguido pelo número 2, indicando que devemos digitar a próxima linha, e assim sucessivamente, até que seja digitada a palavra **FIM**, com todos os caracteres em maiúsculas, ou quando o número de linhas exceder a 100 — limite estabelecido pelo programa.

Neste momento o programa passa para o modo "impressão", limpando a tela e imprimindo o texto no formulário, segundo a formatação pré-estabelecida pelo programa.

O MODO DE CORREÇÃO

Em qualquer momento da digitação, podemos entrar no modo de correção para corrigir um erro que porventura tenha sido cometido ou se simplesmente desejarmos ver uma linha qualquer, já digitada anteriormente.

Para entrar no ***modo de correção,*** devemos pressionar a tecla *ENTER* na linha que estivermos digitando. Quando o computador mostrar o símbolo de "#" seguido pelo número da próxima linha, devemos digitar o símbolo "#" seguido pelo número da linha que desejarmos — que pode ser qualquer uma —, inclusive a última linha digitada.

Uma vez feito isso, o computador nos apresenta duas possibilidades de escolha: digitar novamente a linha ou corrigí-la.

Se escolhermos a primeira opção, basta, após indicar a escolha ao computador, digitar a linha novamente, lembrando-se que existe um limite ao número máximo de caracteres.

Na segunda opção teremos à nossa disposição, vários caracteres auxiliares com funções de correção:

Espaço — a tecla de espaço permite, ao ser pressionada, mostrar a linha caractere por caractere.

L — a letra L maiúscula permite visualizar-se a linha toda na tela.

B — tem a função de **backspace,** ou seja, retorna o cursor sem apagar o caractere mostrado no vídeo.

D — apaga o caractere imediatamente à direita.

I — permite a inclusão, em qualquer ponto da linha de vários caracteres. Uma vez digitado o caractere "I", os caracteres usados como comando, passarão a atuar como caracteres normais.

E — volta ao modo de introdução de linhas, na mesma linha onde foi abandonado.

Sempre que a linha estiver muito grande, o computador avisará, exibindo uma mensagem e retornando ao modo de correção.

POSSIBILIDADES DE MUDANÇAS

Este programa poderá ser modificado de acordo com as suas necessidades e sua imaginação. Por exemplo, podemos eliminar as linhas relativas à correspondência e torná-lo mais geral. Poderemos, ainda, tentar transformar a rotina de verificação do tamanho da linha (linha 240) em uma rotina que, além de verificar o tamanho da linha digitada, "quebra" o seu texto, transferindo os caracteres excedentes para a linha posterior.

As modificações mais interessantes, no entanto, são as que tornam este programa compatível com o TK. Mostraremos, na próxima parte, um conjunto de dicas para traduzirmos este programa para o seu computador. Estas dicas são interessantes mesmo para quem não vá fazer um programa de edição de cartas, pois elas valem para vários outros programas.

Nota: A listagem que mostramos não imprime o texto, mas mostra-o na tela. Isso é conveniente na fase de ***debug,*** quando eliminamos os erros de lógica e de digitação, pois evita-se o gasto de papel desnecessário. Uma vez corrigido o texto, mude as instruções PRINT das linhas 260 à 430 para **LPRINT.** ○

RESPOSTA DO QUEBRA-CABEÇA

Motoqueiro no Deserto

Samuel Ejchel.

Dentre as muitas respostas que recebemos de nossos leitores do Quebra-Cabeça "Motoqueiro no Deserto" encontramos, mais uma vez, a de **Samuel Ejchel.**

Desta vez, porém, ela nos chegou às mãos a tempo de ser escolhida para publicação. Isso pela clareza e simplicidade com que ele abordou e resolveu o problema.

Recebemos muitas outras respostas de nossos assinantes, porém estavam corretas apenas as enviadas por

André B. Hartmann
Antonio Cesar Lettieri
Álbio Böing.
Marcos Antonio Pichateber

O Quebra-Cabeça da revista nº 3 admite muitas soluções ou instruções equivalentes, que conduzem sempre a uma distância máxima de 275 km.

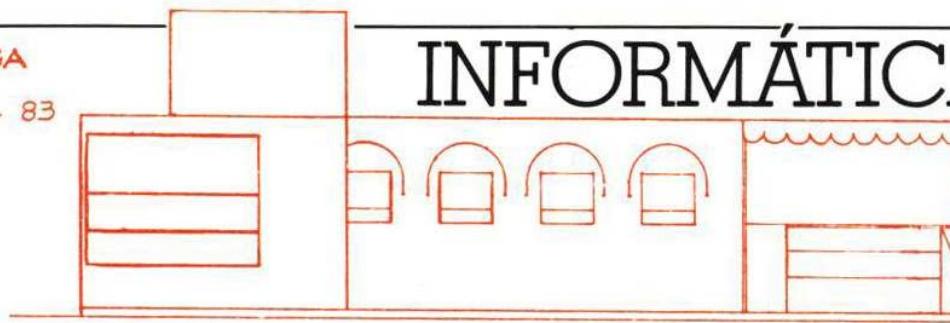
O itinerário na condição da solução ótima, pode ser decomposto em 3 trechos:

1. O último, que o motoqueiro percorre apenas duas vezes, uma indo outra voltando. Para ser máximo, ele deve partir com tanque totalmente cheio e caminhar 150 km para frente e em seguida retornar.
2. Como existe combustível disponível para 900 km e o tanque da moto leva o equivalente a 300 km, haverá um primeiro trecho que, no mínimo, deverá ser percorrido 6 vezes, 3 indo e 3 voltando. Isto porque 900:300 = 3.

Para otimizar, este trecho deve ser o mais curto possível.

3. No segundo trecho, que liga o fim do 1º com o início do último, deve-se levar e deixar combustível suficiente para os 300 km do último trecho. Como a capacidade total do tanque é equivalente a 300 km não será possível fazer isso em menos que 2 etapas, ou seja, a moto deverá percorrer o segundo trecho no mínimo 4 vezes, duas indo e duas voltando. Como no primeiro a moto percorre 6 vezes e neste só 4, devemos fazer este trecho o mais comprido possível, em detrimento do primeiro, ou seja, a moto deve sempre partir com o tanque cheio e caminhar a maior distância possível, deixando sobrar no fim o equivalente a 150 km por etapa. Assim este trecho deverá ter 75 km de forma que a moto sai com o equivalente a 300 km (tanque cheio), viaja 75 km, deixa o equivalente a 150 km, e retorna consumindo o equivalente a 75 km restantes no tanque. O consumo total neste trecho será também do equivalente a 300 km (4 x 75 km).
4. Finalmente, para o primeiro trecho, resta combustível equivalente a 300 km, ou seja, 900 iniciais menos 300 consumidos no 2º trecho e menos 300 consumidos no último. Ora, como ele é percorrido 6 vezes, sua extensão será de 50 km (300:6). Com isso a distância máxima atingida pela moto seria 50 + 75 + 150 = 275 km.
5. Entre as várias alternativas possíveis de programa, a mais simples parece-nos a seguinte:

```
 10 ENCHA O TANQUE NA ALDEIA.
 20 PENETRE 50 KM NO DESERTO.
 30 FACA UM  RESERVATORIO COM
    COMBUSTIVEL PARA 200 KM.
 40 RETORNE PARA A ALDEIA.
 50 REPITA AS INSTRUCOES 10 A
    40 MAIS UMA VEZ.
 60 ENCHA O TANQUE  NA ALDEIA
    PELA TERCEIRA VEZ.
 70 PENETRE  NO  DESERTO ATE"
    ATINGIR OS  RESERVATORIOS
    A 50 KM DA ALDEIA.
 80 COMPLETE O TANQUE DA MOTO
    RETIRANDO  COMBUSTIVEL DE
    UM DOS RESERVATORIOS.
 90 PENETRE MAIS 75 KM NO DE-
    SERTO.
100 FACA UM  RESERVATORIO COM
    COMBUSTIVEL PARA 150 KM.
110 RETORNE ATE OS RESERVATO-
    RIOS A  50 KM DA ALDEIA.
120 ENCHA O  TANQUE DA  MOTO.
    DEVEM TER RESTADO COMBUS-
    TIVEL SUFICIENTE PARA  50
    KM NUM DOS RESERVATORIOS.
130 PENETRE  NOVAMENTE  75 KM
    ATINGINDO O  RESERVATORIO
    COM  COMBUSTIVEL PARA 150
    KM.
140 COMPLETE O  TANQUE DA MO-
    TO.   DEVEM RESTAR NO RE-
    SERVATORIO,   COMBUSTIVEL
    PARA 75 KM.
150 PENETRE MAIS  150  KM  NO
    DESERTO.
160 PLANTE A BANDEIRA.
170 RETORNE ATE" O RESERVATO-
    RIO QUE CONTEM  COMBUSTI-
    VEL PARA 75 KM.
180 COLOQUE NO  TANQUE TODO O
    COMBUSTIVEL EXISTENTE  NO
    RESERVATORIO.
190 RETORNE ATE" O RESERVATO-
    RIO QUE CONTEM  COMBUSTI-
    VEL PARA 50 KM.
200 COLOQUE NO TANQUE  TODO O
    COMBUSTIVEL  EXISTENTE NO
    RESERVATORIO.
210 RETORNE A ALDEIA.
```

O

INFORMÁTICA 83

Ana Lúcia de Alcântara

Nos próximos dias 17 a 23 de outubro, São Paulo será palco de um grandioso evento organizado pela Guaselli e SUCESU, contando com o apoio da ABICOMP: O INFORMÁTICA/83.

Este evento, a nível internacional, conta com a realização de dois acontecimentos importantes, ou seja: a III Feira Internacional de Informática e o XVI Congresso Nacional de Informática, ambos acontecendo paralelamente um ao outro.

Este evento é de tão grande importância que, juntos, são considerados o terceiro maior acontecimento no mundo, na área de Informática, igualando-se ou, superando até, o NCC nos EUA e o SICOB na França.

A programação do INFORMÁTICA/83, foi estruturada toda ela na preocupação em atingir não somente aos profissionais e fabricantes do setor, mas também ao jovem e a criança — que serão os "amantes da informática no futuro" —, como também aos leigos, ou seja, aqueles que não possuem um conhecimento mais aprofundado da área. "Uma oportunidade para que todos aqueles que se interessam pelo fantástico mundo da informática, tenham acesso às informações básicas" — esclareceu Salvador Perroti, presidente do Evento INFO/83.

Esta programação terá uma movimentação calculada em cerca de 200 mil pessoas, conforme expôs o presidente do Evento (Salvador Perroti) —, com uma área ocupada, na Feira, de cerca de 23 mil metros quadrados, sendo que 19 mil será ocupada por expositores (mais ou menos 70 por cento constituída de equipamentos fabricados no Brasil, de empresas com desenvolvimento e tecnologia próprios); dois mil metros ocupados por universidades brasileiras e mil por entidades e instituições governamentais.

Assim sendo, na opinião do presidente da Feira José Roberto Faria Lima, "a grande vedete deste evento, será a presença marcante da tecnologia nacional, e, "na feira", (acrescentou José Roberto), "será a área de micros, pois esta é a área mais próxima do grande público — a área de contato da informática com o público que hoje existe no Brasil".

A Feira:

A III Feira Internacional de Informática na realidade, será a "chamariz" para toda a programação do evento. Com inúmeras atrações: atividades que incluem debates para universitários, para o grande público e para empresários e profissionais de informática, abordando temas centrados no universo da informática, não só a nível de desenvolvimento da área como avaliação da tecnologia adotada pelo Brasil; além de temas referentes a política nacional de software e também referente às preocupações de todos com a regulamentação da profissão e o sindicalismo na área; finalizando esta série de debates na feira, estará presente a preocupação com a análise da presença do Estado na economia da área de informática.

Além destes debates haverá também um circo montado especialmente para as crianças, onde estas poderão tomar contato com os micros que estarão no picadeiro do circo, através de orientadoras e pedagogos especialmente treinados para "brincar" com as crianças e o computador. E um robô, trazido especialmente para a feira, jogará uma partida de xadrez com o vencedor de um jogo realizado numa praça pública com um tablado e com peças gigantes.

Outra preocupação dos organizadores desta feira, foi com a melhor forma de locomoção do público dentro do Anhembi. Para isto, os stands foram distribuídos por área, mostrando os produtos nacionais e estrangeiros lado a lado. Para a orientação do público, foram pintadas, no chão, setas coloridas indicando o circuito a ser percorrido.

O Congresso

O XVI Congresso Nacional de Informática tem uma tradição que remonta dos tempos difíceis de ano de 1967 que, na época, foi chamado de Congresso Nacional de Processamento de Dados e que reuniu pouco mais de 500 pessoas.

Hoje em dia o Congresso tem o objetivo de colocar as técnicas automatizadas não só aos técnicos, como também ao grande público.

Os seminários e debates do Congresso, contarão com a presença de expositores de renome internacional como por exemplo Ralph Gomory falando sobre a Evolução da Tecnologia dos Computadores; John Maccarthy abordando o tema "Inteligência Artificial" e Y.F. Lum falando sobre Videotexto, além de outros. Os temas versarão sobre diversos assuntos, tentando sempre atingir a todos os interesses. No dia 17, haverá a abertura, com seção solene e o congresso se estenderá até o dia 23 as 17 horas.

O Congresso estará distribuído no Anhembi, desde o plenário, ampliando-se aos auditórios E, G, J, na sala A, B, H e L; iniciando-se as palestras, debates e cursos técnicos às 8 horas e encerrando-se às 17 horas.

O intuito real dos organizadores de tamanho evento é transmitir a todos os frequentadores, a imagem de que a convivência entre o homem e o computador pode ser pacífica e recheada de trocas de informações úteis ao desenvolvimento da humanidade. Tanto é que, na feira, haverá uma exposição de "artes plásticas", reunindo painéis, pinturas, gráficos e desenhos de vários autores feitos por computadores em todo o país.

Diante de tudo isto onde estará a nossa revista?

Nosso stand será especial; numa esquina localizada na rua M, número 7 no formato de uma banca de jornal, com computadores TK funcionando durante todos os sete dias da feira.

ARTIGOS

PROTEÇÃO JURÍDICA DO PROGRAMA DE UM COMPUTADOR

1ª parte

Eduardo J. Vieira Manso / Advogado

A questão da protegibilidade, ou não, do programa de computador eletrônico, através de normas jurídicas adaptadas ou próprias, não é muito novo. Em 1964, o ***Copyright Office,*** que é o órgão da Biblioteca do Congresso Norte-americano encarregado do desempenho de todas as funções e obrigações administrativas decorrentes da legislação sobre o direito autoral estadunidense (o chamado ***copyright***), já havia decidido efetuar o registro — como sendo obra suscetível daquela proteção — dos programas de computador eletrônico, desde que (a) fossem o resultado de uma autoria pessoal (contendo, portanto, originalidade), (b) já tivessem sido publicados (isto é, postos à disposição do público) e (c) fossem entregues (àquele órgão) cópias elaboradas de maneira humanente legíveis. Em 1967, o Centro de Estudos Internacionais da Propriedade Industrial, que é órgão da Faculdade de Direito e de Ciências Políticas e Econômicas de Estrasburgo, na França, realizou um colóquio de engenheiros de processamento de dados, juristas e outros cientistas, sob o tema "A PROTEÇÃO DOS RESULTADOS DA PESQUISA EM FACE DA EVOLUÇÃO DAS CIÊNCIAS E DAS TÉCNICAS", quando foram examinadas questões atinentes à Microbiologia, à Química Macrocelular, à Informática e, em particular, aos Programas de Computadores Eletrônicos. Se bem que, naquela altura, o Parlamento Francês já discutia o projeto de lei (promulgada em janeiro de 1968) que excluiu da proteção pela Propriedade Industrial os programas, negando-lhes os direitos de patentes de invenção e os de privilégios correspondentes, houve grande debate quanto à patenteabilidade dos programas, defendida por maioria. A tendência naquele colóquio foi no sentido de que os programas poderiam ser protegidos pelo direito decorrente da concessão de patentes de invenção. Porém também houve juristas que defenderam a tese de que o programa poderia e deveria ser protegido pelas regras do Direito Autoral, tendo em vista tratar-se de verdadeira obra do espírito humano, de natureza evidentemente intelectual, sujeita à reprodução e execução, tal como se dá com os roteiros cinematográficos, ou com os projetos de engenharia (posição até hoje defendida pela maioria dos doutrinadores italianos, em razão de peculiaridade do seu direito autoral interno, como se verá mais adiante). Além do exemplo norte-americano, também as Filipinas introduziram, expressamente, no seu Decreto de 14 de novembro de 1972 (relativo à propriedade intelectual), os programas de computadores, como obras às quais ele se aplica, segundo dispõe o seu art. 2, na alínea ***"n"***.

No Brasil, JOSÉ CARLOS TINOCO SOARES parece ter sido o pioneiro na publicação de estudo a respeito do assunto, quando divulgou um artigo intitulado ***PROTEÇÃO DOS PROGRAMAS DE COMPUTADORES,*** através da Revista de Direito Mercantil, volume 17, de 1975 nas páginas 39/44. Ali, TINOCO SOARES defendeu a inaplicabilidade do Direito Autoral aos programas, assim como das regras atuais da Propriedade Industrial (patente de invenção, ou privilégio de modelo industrial), nada obstante tivesse proposto a introdução, em nosso sistema de direito positivo, de um ***privilégio dos programas de computadores, por um tempo determinado e não excedendo o máximo de cinco anos*** (ob. cit. pág. 44).

Posteriormente, ROBERTO FRANK, CLÁUDIO DE SOUZA AMARAL e HENRIQUE GANDELMAN trataram do assunto em artigos publicados no periódico TELECOM (de 09.11.79 e 06.08.79, os dois primeiros, respectivamente) e no JORNAL DO BRASIL de 24.08.80, o último, todos defendendo, no entanto, a inclusão dos programas entre as obras protegidas pelo Direito Autoral, embora com alguma temperança.

Em 1979, a **CAPRE** — Comissão de Coordenação das Atividades de Processamento Eletrônico —, extinta com a criação da SECRETARIA ESPECIAL DE INFORMÁTICA — SEI —, que a substituiu, organizou um Grupo de Trabalho sobre Mecanismos de Controle à Importação de ***Software,*** em cujo relatório propôs, entre outras sugestões, que, "no Brasil, se tomem as primeiras medidas para o estabelecimento de uma estrutura legal e jurídica adequada, sem optar pela legislação de direito autoral ou pelo direito de patente, que procure preservar e estimular as empresas nacionais do ramo e aumentar em quantidade e qualidade a oferta de empregos para os profissionais brasileiros".

Essa parece ser, mesmo, a tendência geral, a despeito dos Estados Unidos da América, cuja Lei nº 96-517, de 12.12.80, alterou a legislação relativa aos direitos autorais, para deixar inequívoca sua proteção, como tal, aos programas de computadores eletrônicos, agora expressamente tratados como obras intelectuais geradoras do ***copyright,*** ratificando assim, como é de costume por lá, a posição antes assumida pelo ***Copyright Office.***

A **OMPI** — Organização Mundial da Propriedade Intelectual —, entidade intergovernamental e órgão especializado da UNESCO, a qual tem sede em Genebra, e cujo principal papel é promover a proteção da propriedade intelectual (entendida esta como os direitos autorais e a propriedade industrial) por toda a Terra, além de administrar algumas das mais importantes convenções multilaterais relativas ao Direito Intelectual —, depois de ouvir especialistas de diversos países a ela filiados, criou, também, um Grupo Consultor de peritos governamentais, a fim de avaliar a necessidade da proteção dos programas e a natureza de tal proteção. Entre outros motivos justificadores da oportunidade de uma proteção jurídica, segundo o representante da Alemanha Ocidental, foi ressaltado o fato de que, se não houver nenhuma proteção, os interessados em obtê-la haveriam de adotar medidas de segredo (co-

mo, aliás, algumas empresas produtoras de equipamentos fazem), afirmando que isso *não é desejável, do ponto de vista do progresso dos conhecimentos técnicos.*

Depois de diversos anos de reuniões, estudos e debates, a OMPI aprovou e publicou um conjunto de regras que ofereceu aos governos dos países filiados, para servirem de guias, modelos ou sugestões para a elaboração de suas próprias legislações internas. A tais regras foi dado o título oficial de "Disposições Tipos" e elas consagram a introdução, no mundo jurídico-legal, de um novo e especial direito para a proteção dos programas de computadores eletrônicos, que, também, participa da classificação dos Direitos Intelectuais (ou da Propriedade Intelectual, como também são chamados aqueles direitos).

Além disso, a **OMPI** também concluiu pela oportunidade de propor aos governos dos países a ela filiados a aprovação de um tratado internacional sobre a proteção do LOGICIÁRIO (palavra que é por mim proposta para substituir *software*, seguindo o exemplo dos franceses que a trocaram por *logiciel*).

De um modo geral, as disposições tipos têm a seguinte estrutura (que se transcreve da revista *Le Droit d'Auteur*, de 1978, pág. 13): *O artigo primeiro define o objeto da proteção ("programa de computador", "descrição do programa", "documentação auxiliar" e "logiciário", este último consistindo em um ou vários dos objetos mencionados anteriormente), assim como define o termo "proprietário". O artigo 2 determina a quem pertencem os direitos relativos ao logiciário, particularmente quando este foi criado por um empregado; além disso, regulamenta a transferência e a reversão dos direitos relativos ao logiciário. O artigo 3 define a exigência de originalidade do logiciário. O artigo 4 precisa que as noções (por oposição à forma em que elas são expressas) ficam fora dos limites da proteção fixada pela lei. O artigo 5 enumera os atos cobertos pelos direitos do proprietário, cuja lista pode dividir-se em duas partes: os pontos I e II tratam da divulgação não autorizada do logiciário e do acesso não autorizado ao logiciário, enquanto os pontos III e IV são concernentes aos atos de cópia, de utilização, de venda, etc, não autorizados, do logiciário. O artigo 6 define a violação e prevê dois casos que não são considerados como violações (criação independente do logiciário e a situação especial dos navios, dos engenhos espaciais ou engenhos de locomoção aérea ou terrestre estrangeiros que penetram o território de um país). O artigo 7 define a duração dos direitos conferidos pela lei* (20 anos, contados da data da criação do logiciário). *O artigo 8 define as medidas judiciais disponíveis em caso de infração. O artigo 9 precisa que uma proteção fundada em outras disposições (legais) não fica excluída.*

Em junho do corrente ano, a mesma OMPI preparou um texto de um "Esboço de Tratado para a Proteção do Logiciário" (em inglês: *Draft Treaty for the Protection os Computer Software)*, cuja estrutura jurídica é basicamente a mesma daquela que têm as Disposições Tipos, nas quais foram, evidentemente, calcados seus artigos. Assim, os seis primeiros artigos do "Esboço de Tratado" referem-se ao direito relativo à proteção do logiciário: definição; princípio da proteção (ou seja, prescreve a que a proteção a ser conferida em cada país signitário do tratado deverá ser no mínimo igual à proteção prevista no próprio tratado, o que, todavia, não impedirá que a proteção nacional seja ainda mais ampla que a do tratado); tratamento nacional (isto é, o tratado prescreve que aos estrangeiros, oriundos de países filiados a ele, recebam o mesmo tratamento que é dado aos nacionais, desde que tenham também cumprido as formalidades eventualmente exigidas, seja no seu país de origem, seja no país em que estiver pleiteando a proteção); os atos contra os quais se aplica a proteção; a duração da proteção (prevista em 20 anos, no tratado, tal como nas Disposições Tipos) e os casos especiais dos veículos que ingressem no território nacional, portando programas de seus países de origem. Os demais artigos do projeto referem-se a assuntos administrativos, como as hipóteses e modos de revisão do tratado, as maneiras de ser parte dele e quando entra em vigência, assim como sua denúncia, etc.

Simultaneamente, a **OMPI,** através do seu Comitê de Peritos Governamentais, prepara-se para examinar, também, a questão da protegibilidade dos "circuitos integrados" (também conhecidos

como *chips*), que são alvo de um desenvolvimento significativo, nos últimos anos, principalmente como incremento da fabricação de inúmeros produtos para uso diário, principalmente para comandar certos processos eletrônicos, como os das modernas máquinas de lavar, dos jogos, de brinquedos, etc. A proteção dos circuitos integrados, no entanto, ao menos segundo as regras previstas para a proteção dos programas de computadores, parece bastante incerto e algo remota. Por isso, a **OMPI** preferiu, a conselho dos peritos que integram seu Comitê Intergovernamental, antes de propor qualquer regulamentação protetiva, levantar todas as questões que podem ser suscitadas a respeito, para equacioná-las e verificar se, quando e qual proteção poderá ser sugerida e proposta.

No Brasil, portanto, ainda não existe nenhuma norma legal expressamente destinada a proteger os programas de computadores eletrônicos. A Secretaria Especial de Informática – SEI – está diligente e cuidadosamente elaborando um esboço de anteprojeto de lei, visando a estabelecer a proteção jurídica dos programas de computadores, o qual esta estruturado mais ou menos no molde das Disposições Tipos da **OMPI** (à qual o Brasil é filiado). Com muita cautela e segurança, a **SEI** está submetendo o seu estudo à análise e ao exame dos especialistas brasileiros, pedindo-lhes sugestões, a fim de oferecer ao Governo um anteprojeto atual, moderno e eficaz, cujas normas possam conferir a melhor proteção jurídica aos programas, aos seus autores e, principalmente, procurando prevenir os interesses nacionais a respeito, que hão de estar acima dos interesses privados, tendo-se em conta que o Mundo será de quem detiver o maior número possível de informações e as puder processar eficientemente.

Acompanhando de perto as sugestões formuladas pelo Grupo de Estudos organizado pela **CAPRE,** assim como atendendo, no que lhe foi possível, as recomendações apresentadas pela "Comissão Especial nº 003/***Software*** e Serviços" (CE-003), que a própria SEI constitui, nos termos da Portaria nº 003, de 20.03.80, procura-se instituir um novo direito, especial para atender às necessidades também especiais do seu complexo objeto. Esse direito, no entanto, por sua natureza e em razão principalmente da natureza de seu objeto, também pertence à classificação de Direito Intelectual, situando-se, portanto, ao lado do Direito Autoral e do Direito de Propriedade Industrial, dos quais será o irmão mais novo, posto que, sem nenhuma dúvida, o programa de computador, bem como todo o logiciário, é uma obra intelectual, evidentemente. Assim, não se estarão aplicando, integralmente, as regras do Direito Autoral, como haveria de acontecer se se admitisse, definitivamente, que os programas fossem obras objeto de tais regras, as quais, em conjunto, são perniciosas para o interesse nacional e até mesmo para a própria proteção pretendida. E não se aplicarão as regras do Direito da Propriedade Industrial, que, aliás, estão praticamente vedadas, pelo que dispõe o Código da Propriedade Industrial, cujo artigo 9º, entre as "invenções não privilegiáveis", inclui, no inciso ***h***, "os sistemas e programações, os planos ou os esquemas de escrituração comercial, de cálculos, de financiamento, de crédito, de sorteios, de especulação ou de propaganda".

Contudo, ***enquanto "seu" lobo não vem,*** isto é, na falta de normas jurídicas especialmente aplicáveis aos programas, como poderão, os interessados em obter alguma proteção, defender esses mesmos interesses, principalmente quando ocorrerem atos de terceiros que lhes firam o que a sua consciência lhes aponta como "seus direitos". O que fazer contra a pirataria, que campeia solta?

(Continua no próximo número)

PEQUENOS

ANÚNCIOS

REVISTAS IMPORTADAS — FARAH'S BOOK SHOP — *Rua Haddock Lobo, 1503 — Fone: 881-3410.*

Vendo Timex/Sinclair 1000, ZX SPECTRUM, produtos MEMOTECH (HRG, 64K, Interface para impressora, etc.), programas e outros periféricos. Tratar com TANIOS — *Tel.: 270-3368.*

Serviço de Datilografia — IBM. Teses, apostilas e manuais técnicos. *Telefone: 258-8486 — Américo, SP.*

GRANDE CIRCUITO — Boletim para usuários do TK. Duas mil cópias em Off-set distribuídos para todo o Brasil. Anúncios grátis. Informações: *CP: 28, CEP: 27200 — Piraí, RJ.*

SOFTSYSTEM — Software importado para TK82/85, NEZ, CP 200, ZX 81, etc. Linguagem de máquina inéditas. Solicite catálogo para *Caixa Postal 02, Poá, SP. CEP: 08550.*

Troco ou vendo **PROGRAMAS (NACIONAIS E IMPORTADOS)** para os micros: CP 200, TK82-C e ZX 81. Possuo Simulador de Vôo, Turbo, Galaxy e muitos outros jogos e aplicativos. Tratar com *Eduardo Medeiros — R. Eliseu Guilherme, 1076 — Ribeirão Preto — SP — CEP: 14100* — Atendo também por reembolso.

Vendo ou troco **PROGRAMAS DE JOGOS** para TK82-C e TK85. Tratar com *Edson — Rua Guariba, 54 — São Paulo. Tel.: 918-0998 (c/ Ruth).*

RADIO MICRO — O primeiro grupo de rádio amadores digitais do Brasil convida todos os radioamadores a conhecerem os projetos de **hard** e **soft** que estamos desenvolvendo. Aos interessados entrarem em contato com *PY-2-EMI Renato Strauss — R. Cardoso de Almeida, 654/32 — 05013 — São Paulo, SP.*

Colocamos **VIDEO-REVERSO** em televisores TX-Philips, fazemos adaptações de TV's para monitor de vídeo e vendemos kits de vídeo-reverso. *Luiz Wellington — Tel.: 224-2776 — Fortaleza/CE.*

Vendo programas nacionais e importados, 2K e 16K, BASIC e ASSEMBLER para micros TK82-C e similares (Aceito trocas) — *Carlos A. Sciaretti — Caixa Postal 5567 — São Paulo, SP — CEP 01051 — Tel.: 522-8586.*

Conversor do NEZ8000 para SLOW: Em São Paulo, *WILSON DE ASSIS* faz a inclusão desta função no NEZ8000 por Cr$ 20.000,00 (mais as despesas de embalagem e correio) e dá a garantia de 6 meses. Se o seu endereço for próximo ao dele, ele entregará em casa. O seu telefone é: *(011) 203-7967* e o endereço: *Rua Fabrício Correia, 145, Tucuruvi, 02311 — São Paulo — SP.*

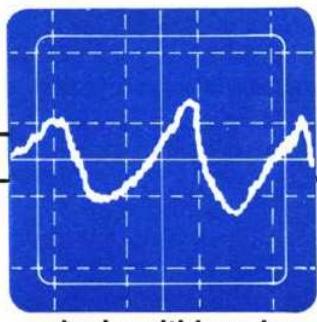
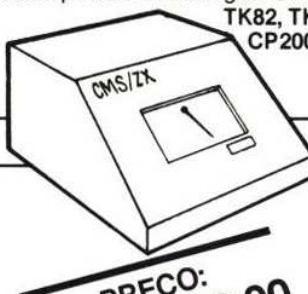
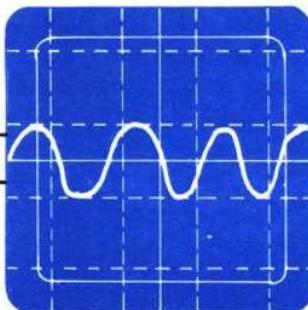
Vendo **esquema completo** do microcomputador **NEZ8000** (com SLOW e expansão de memória) — 3 cópias heliográficas. Cheque nominal de Cr$ 6.000,00 para *JAN MARTIN LUND, Rua Frederico Ozanã, 16/21 — 11100 — Santos — SP.*

Vendo ou troco **programas de jogos para TK82-C** e TK85. Tratar com *Edson, R. Guariba, 54 — São Paulo — Tel.: 918-0998 (c/ Ruth).*

Curso de Linguagem de máquina para TK82 e TK85 (ASSEMBLY Z80). Aos sábados das 9:00 às 13:00 hs. *— São Paulo — Fone: 813-4555 (Márcia) à tarde.*

Troca de programas para TK — *Carlos Horácio C. Fontenelle — R. Alfeu Aboim, 770 — Aldeota — Fortaleza — CE — CEP: 60000.*

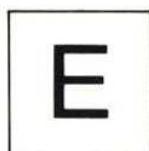
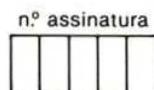
Sou possuidor de muitos **programas para o TK82.** Os interessados em trocá-los escrevam para *Marco Aurélio Dias de Oliveira — Av. Afonso Pena, 1557 — Bloco B — apto. 214 — Campo Grande/MS — CEP: 79100.*

ESPIÃO — Jogo de Aventura para TK e CP-200. Um desafio a sua Memória e Habilidade. 16K com SLOW. Cr$ 5.000,00. Pedidos pelo reembolso para *Cx. Postal: 28 — 27200 — Piraí — RJ.*

0 1 2 3 4 5 6 7 8 9

micromega

MICROMEGA Publicações e Material Didático Ltda.
Caixa Postal 60081 - CEP 05096 - São Paulo - SP
CGC.: 52.275.724/0001-41 - Inscr. Est.: 110.862.362

n.º assinatura

E

Desejo assinar a Revista MICROHOBBY (12 edições)

NOME

ENDEREÇO

TELEFONE

CGC ou CPF

INSCRIÇÃO

CEP

CIDADE

EST.

ASSINATURA

Estou anexando a quantia de Cr$ ______ ☐ em cheque nominal
para MICROMEGA P.M.D. Ltda., referente a venda de uma assinatura. ☐ vale postal
em cheque nominal n.º ______ Banco ______
Data ___/___/___

ESTE CARTÃO NÃO PODE SER UTILIZADO POR TERCEIROS PARA COLETA DE ASSINATURAS
Válido somente se postado diretamente pelo assinante.

COMO COLABORAR COM MICROHOBBY

A Revista MICROHOBBY foi criada para servir de intercâmbio entre os leitores que participam do mágico mundo da computação.

A característica realmente inovadora do computador pessoal, está em transformar cada consumidor num criador. Aproveite sua criatividade e envie suas colaborações recebendo remuneração a título de DIREITO AUTORAL.

A maneira ideal de nos enviar o material a ser publicado obedece às seguintes normas:

1. Nunca esqueça de colocar o nome completo, telefone, endereço e número de sua assinatura em **todo** material enviada a nós, sejam listagens de impressora, fitas, envelope, carta ou qualquer outro material.
2. Envie a listagem de programa **datilografada** ou, melhor ainda, tirada na impressora do computador.
3. Coloque sempre uma linha REM com o nome do autor e o título do programa.
4. Envie uma fita com o programa gravado **algumas vezes** (se possível em gravadores diferentes).
5. Na fita, gravar **com microfone** (em viva voz), algumas instruções úteis:
nome completo e **endereço do remetente.**
6. Quando o programa for adaptado e/ou traduzido de outra revista, citar a fonte (autor original, data de publicação, nome da revista e todos os detalhes que houver referente à publicação).
7. Anexar ao material, uma carta autorizando a publicação por parte da revista e assumindo a responsabilidade pela autoria do material e/ou adaptações. Nesta carta, para agilizar a remuneração, podem constar os dados da conta corrente onde daremos o depósito correspondente aos direitos autorais.
8. O material **não utilizado não será devolvido,** ficando a critério da redação a decisão final sobre sua publicação.
9. O material deve ser enviado para:

MICROMEGA PUBLICAÇÕES E MATERIAL DIDÁTICO
SEÇÃO PROGRAMAS DO LEITOR
RUA BAHIA, 1049
01244 – SÃO PAULO, SP.

COMO FAZER SUA ASSINATURA

A nossa revista não é distribuída nas bancas. Para obter seu exemplar mensal, contendo muitos programas para o seu TK, além de muitas dicas e prêmios interessantíssimos, você deve fazer uma assinatura de **Micro-Hobby** num preço anual de Cr$ 11.800,00.

Para tanto, você deverá preencher corretamente o cupom anexo, colocá-lo num envelope, junto a um cheque nominal ou vale postal a favor de MICROMEGA PUBLICAÇÕES E MATERIAL DIDÁTICO LTDA., no valor de Cr$ 11.800,00 (até 30/10).

Qualquer reclamação deve ser feita por carta anexando, se possível, uma etiqueta de endereçamento, para que o atendimento seja o mais rápido possível.

O envelope deverá ser selado e endereçado à:

MICROMEGA P.M.D. LTDA.
DEPTO. DE ASSINATURAS

Caixa Postal 60081 – CEP 05096
São Paulo – SP

No verso do cheque escreva:

"Destina-se ao pagamento de uma assinatura (12 números) da revista MICROHOBBY".

Quando este cheque for devolvido ao seu Banco com nosso endôsso, servirá de comprovante provisório até que nosso recibo seja enviado pelo correio.